MICRO MAG

LISTING BIDOUILLE PROGRAMMATION

CPC
- «Death's Ticket», shoot'em up d'enfer !

ST
- Objets fractals & récursivité

AMIGA
- «Animatix», donnez vie à vos sprites

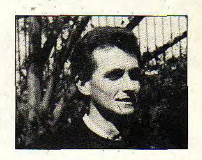
PC
- Manette des gaz pour simulateurs de vol



N° 14 MENSUEL JUILLET/AOUT 1990 - 25 F
BELGIQUE 180 FB-SUISSE 7,50 FS - CANADA 6,75 $CAN

SOMMAIRE JUILLET AOUT 1990 N°14

EDITO



Si vous avez le choix, ne partez pas en vacances.

N'apportez pas votre tribut au saccage du littoral. Evitez les joies illusoires d'un brusque dépaysement. Réprimez vos envies, sources de dépenses somptuaires. La drague est risquée, le soleil malsain, les plages polluées et l'ambiance estivale sujette à caution.

Dédaignez à juste titre l'acquis du front populaire, ou plutôt, usez-en à rentabiliser votre matériel micro.

Œuvrez à nous fournir dès la rentrée, de quoi alimenter nos chères rubriques...

Jean-Claude Paulin, Tahiti.

Directeur de la publication: Jean Kaminsky.

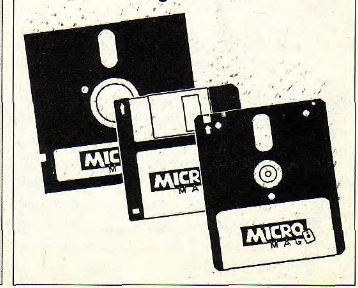
REDACTION
Rédacteur en chef de ce numéro: Jean-Claude Paulin.
Ont collaboré à ce numéro: David Farenzena, Claude Le Moullec, Daniel Provenier, Stéphane Rodriguez, Sébastien Royer, Jean-François Six, Jean-Yves Trétout.
Couverture: Claude Marrel.
Maquettistes: Jean-Claude Paulin (PAO), Jean-jacques Galmiche.

ADMINISTRATION
Abonnements: Laser Presse OGP - 175, av. Jean-Jaurès, 75019 Paris. Tél. : (1) 42 41 30 10 de 8h 30 à 18h 00 du lundi au vendredi.
Comptabilité: Sylvie Kaminsky.

REGIE PUBLICITAIRE
NEO-MEDIA, 5-7, rue de l'Amiral Courbet, 94160 Saint-Mandé. Tél. : (1) 43 98 22 22.
Chef de publicité: Pascale Kittel.

MICRO-MAG
est édité par Laser Presse SA, 5-7, rue de l'Amiral Courbet, 94160 Saint-Mandé.
Commision paritaire: n°71178.
Dépôt légal: 3e trimestre 1990.
Impression: RBI.

Consultez notre Logithèque Page n°66



Télématique tac toc !

MINITEL ET TÉLÉCHARGEMENT

Reliée à tout ordinateur, une petite merveille nommée Minitel permet d'accéder à de nombreux serveurs et à leurs immenses banques de programmes. Bon voyage au pays du téléchargement...

Le matériel de base est simple, vu qu'il ne nécessite qu'un ordinateur muni d'une prise RS 232, un minitel (1b) et le câble de liaison entre les deux. Pour la partie logicielle, il convient de se procurer un soft permettant la communication entre l'ordinateur et le serveur de votre choix. En effet, les différents serveurs n'utilisent pas tous le même protocole. Le matériel réuni, détaillons-en le fonctionnememt (Les personnes réfractaires aux beautés de la technique moderne, peuvent se référer directement à l'encadré en fin d'article).

La RS232

Elle est une méthode de transmission qui permet d'échanger des informations de façon fiable, avec seulement 3 fils qui sont:

- Txd pour l'émission.
- Rxd pour la réception.
- Une masse (0V) pour servir de référence.

Le standard RS 232 complet comporte une quinzaine de fils avec des timings précis qu'il serait trop long de décrire en détail. Dans le cas d'une liaison ordinateur-minitel, les signaux autres que Txd, Rxd et Ov, sont fixés à certaines valeurs assurant le bon fonctionnement de l'ensemble. Une question se pose: comment faire transiter des octets comportant 8 bits par un seul fil? La solution consiste à les envoyer les uns après les autres, en série. C'est pourquoi la RS 232 est parfois nommée «prise série». Mais ce n'est pas le tout de transmettre, encore faut-il pouvoir effectuer un minimum de contrôles. C'est pourquoi la méthode finalement adoptée est la suivante :

- 1 ou 2 bits de début.
- Le nombre de bits de données (7 pour de l'ASCII, 8 pour du binaire pur).
- 1 bit de parité paire ou impaire.
- 1, 1.5 ou 2 bit de stop.

Il ne reste plus qu'à fixer une vitesse de transmission, en bauds ou bits/seconde. Les échanges entre l'ordinateur et le minitel se font à 1200 bauds. Heureusement, la quasi-totalité des programmes de téléchargement liés à un serveur particulier prennent en charge la configuration de la RS232 et il n'y a pas à s'en préoccuper. En considérant qu'il faut transmettre une dizaine de bits par octet, on obtient, à 1200 bauds, une vitesse de 120 octets/seconde. Soit environ 7 Ko/minute.

Le minitel

L'intérêt essentiel du minitel est d'inclure une carte MODEM qui permet de faire transiter des signaux informatiques par les lignes téléphoniques. En effet, les signaux carrés, générés par la RS 232 de l'ordinateur, seraient complètement déformés à cause de la faible bande passante du réseau. Pendant un téléchargement, le minitel fait seulement office de transmetteur. Pour jouer ce rôle, il utilise le standart V23, c'est-à-dire qu'il émet vers le serveur à 75 bauds et reçoit à 1200. En revanche, la liaison avec l'ordinateur se fait à une vitesse fixe de 1200 bds dans les deux sens. Les minitels 1b ont un MODEM retournable. Dans ce mode, ils peuvent émettre à 1200 bds et recevoir à 75. C'est très pratique quand on veut envoyer des programmes (*upload*) au lieu de recevoir (*download*).

Le câble

Sa fonction est bien entendu de relier le minitel à l'ordinateur (ou le contraire). Mais il joue aussi un autre rôle. Le voltage des signaux émis par une RS232 classique n'est pas compatible avec ce qu'attend la prise péri-informatique du minitel. On trouve donc, dans le capot de la prise RS232, un petit montage destiné à adapter les tension. Le branchement ne présente aucune difficulté si ce n'est que la prise minitel est parfois dissimulée par un petit volet. A noter qu'un tel câble permet également de s'envoyer des programmes entre copains. Il suffit que l'un des deux se transforme en serveur... Pour ce faire, il n'y a qu'à taper «fnct m r» sur un minitel 1b qui retourne alors son MODEM. Ensuite, un bon programme de *comm.* et c'est tout...

Le logiciel

Comme d'habitude, le soft est une composante fondamentale. C'est lui qui établit la synchronisation du téléchargement entre le serveur et votre ordinateur. Il veille également à son bon déroulement en faisant des vérifications régulières. A cet effet, les données sont découpées en paquets dont chacun fait l'objet d'un contrôle. Le soft peut ainsi demander au serveur la réémission d'un bloc déficient. Le mode d'emploi est très simple, il suffit de mettre le soft en attente de données. A l'aide du minitel, on se promè-

ne dans le serveur jusqu'à être séduit par un logiciel quelconque. On ordonne alors le téléchargement, le soft de communication le détecte, enregistre et sauve sur disque. C'est tout!

Compactage et décompactage...

...sont les deux mammelles du téléchargement. En effet, il est important de limiter au maximum le temps de connexion et c'est ainsi que la plupart des programmes présents sur un serveur sont compactés. Par exemple, sur Micromag (le serveur), ils portent alors le suffixe «.ARC», du nom du célèbre programme d'archivage utilisé. Une fois que vous avez téléchargé un «.ARC», il faut lancer le programme de décompactage présent sur la disquette du kit de téléchargement. Après cette dernière étape, votre logithèque s'est enrichie d'un ou plusieurs nouveaux titres car un seul «.ARC» peut contenir différents fichiers.

Conclusion

Il y a encore des foules de choses à dire sur les activités liées au mariage ordinateur-minitel et ce sujet fera bientôt l'objet d'un dossier complet. Disons seulement que le téléchargement et la possibilité pour tous de communiquer à distance via le minitel, ouvrent de nouveaux espaces d'applications à la micro-informatique, l'empêchant ainsi de se refermer sur elle-même (aïe!). Citons les messageries nationales et internationales, le travail à domicile, la possibilité de diffuser vos logiciels...etc.

J-Y T.

Comment télécharger?

- Relier le câble de liaison entre la prise RS232 (prise série) de votre ordinateur et la prise péri-informatique du minitel.
- Connectez-vous au serveur et mettez le programme de téléchargement en attente.
- Demandez au serveur d'envoyer le logiciel ou les données que vous désirez. Le téléchargement se fait automatiquement.
- Si votre fichier porte le suffixe «.ARC», il faut le décompacter. Lancez alors le programme de décompactage fourni avec votre kit de téléchargement.
- Voilà, c'est tout et ce logiciel n'a coûté que la communication.

SYMBOL & SYMBOL AFTER

LES COURS DU PROFESSEUR ALI GATOR

Tout jeu Basic, pas trop indigent, fait appel aux commandes Symbol et Symbol after. Etes-vous certain d'en tirer le meilleur parti?

Depuis des lustres, vous savez redéfinir un caractère avec Symbol. Le principe est simple: un numéro de caractère, 32 à 255, suivi de 8 nombres de 0 à 255, donnent une nouvelle valeur aux 8 octets réservés en ROM pour chaque caractère. Cet espace se situe, pour ceux que cela intéresse, aux adresses &3800 à &3FFF. La ROM n'étant pas modifiable, l'opération a lieu en RAM. C'est là qu'intervient la commande *Symbol After*.

Symbol After

En fait, il s'agit d'une barrière qui permet de réserver, en haut de la RAM, un espace suffisant pour la redéfinition des caractères. Celle-ci n'est pas nécessaire lorsque les caractères redéfinis ont une valeur supérieure à 240. Effectivement, entre &AB80 et &ABFF, figure une zone permettant le stockage et la redéfinition des caractères 240 à 255. Une commande similaire, *Memory*, dresse elle aussi une barrière entre l'espace réservé au basic et celui dédié aux routines binaires. Il y parfois conflit entre ces deux commandes. Nous verrons plus loin comment y remédier. Maintenant, faites en direct:

```
PRINT HIMEM
```

Suivant les configurations de CPC et des extensions utilisées, vous allez obtenir une valeur que nous appellerons «A». Si vous faites :

```
SYMBOL AFTER 240 : PRINT HIMEM
```

La valeur obtenue sera identique à la précédente. Tout simplement parce que l'espace nécessaire pour ces 16 caractères est déjà réservé dès l'initialisation de l'ordinateur. Espace qui peut être récupéré de la façon suivante.

```
SYMBOL AFTER 255 : PRINT HIMEM
```

La valeur obtenue «B» est supérieure à «A» de 8 x 16=128 octets. Ce n'est pas grand chose, mais cela peut toujours servir. Maintenant, faisons :

```
SYMBOL AFTER 40 : PRINT HIMEM
```

La valeur obtenue «C»; est inférieure à «A» de 8 x 200=1600. 1600 octets nécessaires pour modifier 200 caractères. *Symbol After* sert de barrière mais aussi de recopie de la ROM vers la RAM du jeu de caractères. Pour vous en convaincre; tapez les lignes suivantes:

```
10 SYMBOL AFTER 32 : X=HIMEM+1 : MODE 2
20 CLS : FOR H=1 TO 8 : PRINT BIN$(PEEK(X), 8)
30 X=X+1: NEXT: CALL &BB18 : GOTO 20
```

Apparaissent alors un à un tous les caractères dans leur représentation binaire. A partir de là, toutes les manipulations sont possibles: lettres sens dessus dessous, tronquées, inversées, en italiques, etc. A vous de faire preuve d'imagination. Si vous en manquez, essayez les trois petits programmes de notre exemple.

Improper Argument

Ceux d'entre vous qui tapent régulièrement des listings n'ont sûrement pas échappé à l'Improper Argument qui apparaît parfois sur la ligne contenant l'instruction *Symbol After*. Cette ligne, qui en elle-même ne comporte aucune erreur, vient d'entrer en conflit avec une commande *Memory* rencontrée dans le listing. Pour comprendre le phénomène, tapez les lignes suivantes.

```
10 MEMORY &8000
20 SYMBOL AFTER 200
30 PRINT "BONJOUR" : END
```

Quoique l'on fasse, le «Bonjour» de la ligne 30 ne s'affichera pas, le programme renvoyant un têtu *Improper Argument* en ligne 20. *Memory* et *Symbol After* sont deux barrières, mais le système n'admet pas que le *Memory* soit défini avant le *Symbol after*. Il aurait fallu écrire :

```
10 SYMBOL AFTER 200
20 MEMORY &8000
30 PRINT "BONJOUR" : END
```

Dans ce cas, RUN affichera correctement la message, mais faites un second RUN. Coucou, revoilou l'*Improper Argument*! En fait, lors du premier lancement, la barrière *Memory* placée en &8000 reste active jusqu'au *Reset* complet de l'ordinateur. De ce fait, le deuxième RUN recrée la situation du premier programme. Alors que faire? Il existe deux solutions.

- Premier cas: vous travaillez à la mise au point d'un listing dont vous n'êtes pas l'auteur. Le *Memory* reste actif tant qu'un *Reset* n'a pas lieu. Il en est de même pour *Symbol After* et les caractères redéfinis. Donc, lors des nombreux RUN d'essais, inutile de les faire relire par le programme. Rajouter, juste avant ce passage, une petite ligne avec un GOTO pour enjamber l'obstacle. Dans notre exemple cela donnera: 5 GOTO 20. Ainsi, la mise au point pourra s'effectuer sans problème. Celle-ci terminée et avant d'effectuer la sauvegarde définitive, supprimez ce GOTO superflu.
- La deuxième solution est encore plus simple. Il existe une routine système en &BB4E qui réinitialise tout ce qui a trait à la gestion des textes à

l'écran. *Symbol* et *Symbol after* rentrent dans cette catégorie. Ladite routine agit donc comme un Reset sélectif, exactement ce que nous recherchons. Ainsi la ligne : 5 CALL &BB4E appliquée à nos deux exemples, fera disparaître tous nos problèmes d'*Improper Argument*.

Petite restriction: CALL &BB4E positionne le curseur en LOCATE 1,1. En debut de listing, ceci ne devrait pas porter à conséquences.

RECREATION : ALPHA-COLOR

Sur le point de créer un jeu, le mode 1 avec ses 4 couleurs vous semble insuffisant. Il sera donc en mode 0. Or, les 20 caractères possibles par ligne vous posent problème. Comment faire?

Désormais, la solution existe et s'appelle Alpha-color. Ce petit utilitaire permet en effet de créer à volonté un alphabet multicolore autorisant 40 caractères sur une ligne en MODE 0. J'avoue, que l'idée est due à l'équipe *Black System* qui, dans chacun de ses jeux, utilise un tel procédé. J'ai donc conçu un programme basic générant un fichier binaire relogeable et utilisable par tout-un-chacun, même sans connaissance de l'assembleur.

Caractères redéfinis

Tous les chiffres, quelques signes de ponctuation et l'alphabet sont redéfinis deux par deux dans un caractère. Rien ne vous oblige à être d'accord avec mon style de redéfinition. Lorsque vous maîtriserez correctement le logiciel, modifiez à votre guise le dessin des lettres. Car il faut bien l'avouer, certaines comme le N, le M ou le W peuvent prêter à confusion.

Variables de base

Ligne 460 : Memory &3FAF. Une petite routine de capture sera placée entre l'adresse &3FB0 et &3FFF. La zone comprise entre &4000 et &A1E0 est réservée au stockage du fichier où bon vous semble.

Ligne 470 : initialisation d'un DIM pour conserver le numéro des encres.

Lignes 480 - 490 : mode transparent et mode XOR pour sous-programme de recherche des couleurs.

La recherche des couleurs

Il est bien évident que les couleurs de votre logiciel seront préalablement choisies. Ce sous-programme permet d'en reproduire exactement la palette. A noter que je me suis contenté de reprendre un sous-programme déjà utilisé dans «Aux voleurs». Faites de même.

Le questionnaire

La sélection de votre palette de couleurs est la première question à laquelle répondre. Apparaît alors la liste des stylos et les encres qui leur sont affectées. A vous d'en choisir 4 pour rendre votre alphabet multicolore. L'usage m'a prouvé que les mélanges qui en «jettent» sont toujours à base de dégradés. Essayez donc des combinaisons du genre (n° de couleur) 26, 24, 15, 6 ou encore 26, 16, 1, 2 sur fond noir. Résultats garantis.

Reste maintenant à définir l'adresse de départ du fichier binaire qui va se créer. En fait, cela n'a que peut d'importance car il peut être relogé n'importe où. Il suffira de forcer l'adresse lors du chargement par un : LOAD "FICHIER", adresse désirée.

Création du fichier

Ligne 750 : affectation des couleurs.

Lignes 760 - 770 : affichage des lettres redéfinies.

Lignes 780 - 840 : coloration des lettres suivant les stylos choisis.

Lignes 850 - 910 : implantation de la routine d'affichage et capture une à une de toutes les lettres. A ce stade du programme, tout est déjà terminé.

Le fichier

Le fichier ainsi créé mérite quelques commentaires. Il commence à l'adresse que vous avez choisie par une petite routine d'affichage de 32 octets. Celle-ci est calibrée pour des sprites de 2 octets sur 8 lignes; la taille de nos lettres. Après la routine figure la représentation des lettres, le fichier total faisant 720 octets.

J'attire votre attention sur le jeu de caractères disponibles. Vous disposez de tous les chiffres et de tout l'alphabet en majuscule. Ils suivent tous la progression logique telle qu'elle est représentée dans la liste des codes ASCII. Mais entre chiffres et lettres, 7 caractères pour un embryon de ponctuation qui ne correspond pas tout à fait à la réalité. Ainsi, si le «:» est bien à sa place, le signe «<» à été remplacé par le point. Voici les équivalences .

«:» idem.
«;» idem.
«<» remplacé par le point.
«=» remplacé par la virgule.
«>» remplacé par l'apostrophe.
«?» idem.
«à» remplacé par le point d'exclamation.

Utilisation du fichier

Elle est en fait donnée en exemple dans le logiciel. Etudiez avec attention les lignes 920 à 1000.

Ligne 920 : fonction permettant de transformer l'équivalent d'un LOCATE X, Y en MODE 1, en une adresse écran exploitable par la routine d'affichage.

Ligne 930 : data d'une phrase à afficher. Les deux premiers chiffres sont les données X, Y du LOCATE.

Ligne 980 : la phrase est dans la variable A$ en majuscule. Chaque lettre est prise une à une et, dans SP, on calcule son numéro de code ASCII. 48 lui est retiré car le 0 est notre premier caractère contenu dans le fichier.

Ligne 990 : si le caractère n'existe pas dans le fichier, il est remplacé par un espace. Ce dernier, situé à la position 43, est le dernier caractère contenu dans le fichier.

Ligne 1000 : affichage une à une des lettres. La variable ADD est l'adresse de départ du fichier fournie dans le questionnaire. La fonction FN po(ZX, ZY) définie en 920 entre en action. DEP1 = ADD+32. On pointe sur le début de la première lettre. Il suffit de lui additionner la valeur SP multipliée par 16 pour pointer sur la lettre désirée. La première lettre s'affiche, un bouclage affiche toute la phrase. Avec l'espoir que ce petit utilitaire permette de donner un air «pro» à vos prochaines réalisations...

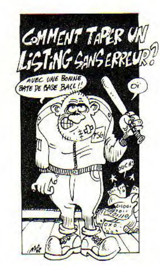
Claude Le Moullec

```
10 REM ::: a l'envers  ::::              [1Ø59]
2Ø MODE 1:SYMBOL AFTER 32:x=HIMEM        [28Ø4]
3Ø FOR h=32 TO 255:FOR g=1 TO 8          [1349]
4Ø va(g)=PEEK(x+g):NEXT g:FOR g=8        [3814]
 TO 1 STEP -1
5Ø z=z+1:POKE x+z,va(g):NEXT:PRIN        [2496]
T CHR$(h);
6Ø x=x+8:z=Ø:NEXT h:CALL &BB18           [2179]
7Ø REM ::: demi caractere ::::           [513]
8Ø MODE Ø:SYMBOL AFTER 48:x=HIMEM        [2774]
9Ø FOR h=48 TO 122:FOR g=1 TO 8          [233Ø]
1ØØ b$=BIN$(PEEK(X+G),8)                 [969]
11Ø a=VAL("&X"+MID$(B$,2,1)+MID$(        [3189]
B$,5,1)+MID$(B$,7,1)+"ØØØØØ")
115 POKE x+g,a                           [3Ø4]
12Ø NEXT:PRINT CHR$(h);:x=x+8:NEX        [2817]
T h:CALL &BB18
13Ø REM ::: caractere nain :::           [583]
14Ø MODE Ø:SYMBOL AFTER 48:x=HIME        [2774]
M
15Ø FOR h=48 TO 122:FOR g=1 TO 8         [233Ø]
16Ø va(g)=PEEK(x+g):NEXT:FOR g=1         [2649]
TO 8
17Ø POKE (x+g),Ø:NEXT                    [1186]
18Ø POKE x+3,va(1):POKE x+4,va(3)        [1552]
:POKE x+5,va(4)
185 POKE x+6,va(5):POKE x+7,va(7)        [1825]
19Ø PRINT CHR$(h);:x=x+8:NEXT h          [1613]
```

```
1Ø REM  ::::::::::::::::::::::::         [1823]
2Ø REM  :                      :         [419]
3Ø REM  :      MICRO MAG et    :         [696]
4Ø REM  :       ALI GATOR      :         [1426]
5Ø REM  :                      :         [419]
6Ø REM  :       presentent     :         [1423]
7Ø REM  :                      :         [419]
8Ø REM  :       ALPHA-COLOR    :         [1189]
9Ø REM  :                      :         [419]
1ØØ REM ::::::::::::::::::::::::         [1823]
11Ø REM :                      :         [419]
12Ø REM :    REDEF ALPHABET    :         [1321]
13Ø REM :                      :         [419]
14Ø REM ::::::::::::::::::::::::         [1823]
15Ø SYMBOL AFTER 199                     [1457]
16Ø SYMBOL 2ØØ,119,119,85,85,118,        [2Ø21]
85,87,Ø
17Ø SYMBOL 2Ø1,118,119,69,69,69,6        [2313]
9,118,Ø
18Ø SYMBOL 2ØØ,114,118,82,82,82,8        [21Ø4]
2,119,Ø
19Ø SYMBOL 2Ø1,119,85,17,17,35,65        [1611]
,119,Ø
2ØØ SYMBOL 2Ø2,23,52,52,83,113,21        [1978]
,23,Ø
21Ø SYMBOL 2Ø3,119,81,65,114,82,8        [2Ø86]
4,116,Ø
22Ø SYMBOL 2Ø4,119,85,85,119,81,8        [255Ø]
1,119,Ø
23Ø SYMBOL 2Ø5,Ø,Ø,Ø,Ø,34,Ø,34,4         [1259]
24Ø SYMBOL 2Ø6,Ø,Ø,Ø,Ø,Ø,34,34,4         [1823]
25Ø SYMBOL 2Ø7,34,37,69,1,2,Ø,2,Ø        [1384]
26Ø SYMBOL 2Ø8,39,39,37,37,39,5,3        [2196]
7,Ø
27Ø SYMBOL 2Ø9,119,119,84,84,1ØØ,        [22Ø2]
84,119,Ø
28Ø SYMBOL 21Ø,1Ø3,119,84,84,86,8        [1798]
4,119
29Ø SYMBOL 211,119,119,69,68,1ØØ,        [18Ø9]
69,71,Ø
3ØØ SYMBOL 212,87,82,82,82,114,82        [221Ø]
,87,Ø
31Ø SYMBOL 213,117,37,37,38,38,37        [1852]
,1Ø1,Ø
32Ø SYMBOL 214,69,71,71,69,69,69,        [2153]
117,Ø
33Ø SYMBOL 215,87,85,117,117,117,        [2471]
85,87,Ø
34Ø SYMBOL 216,119,119,85,85,117,        [1992]
69,71,2
35Ø SYMBOL 217,119,119,84,84,1Ø3,        [2466]
81,87,Ø
36Ø SYMBOL 218,117,117,37,37,37,3        [2248]
7,39,Ø
37Ø SYMBOL 219,85,85,85,85,87,87,        [2314]
37,Ø
38Ø SYMBOL 22Ø,85,85,85,34,34,82,        [2Ø42]
82,Ø
39Ø SYMBOL 221,112,16,16,32,32,64        [1935]
,112,Ø
4ØØ SYMBOL 255,Ø,6Ø,6Ø,6Ø,6Ø,6Ø,6         [4779]
Ø,Ø:SYMBOL 254,126,66,66,66,66,66
,66,126
41Ø REM ::::::::::::::::::::::::         [1736]
42Ø REM :                      :         [419]
43Ø REM :  VARIABLES DE BASE   :         [2Ø81]
44Ø REM :                      :         [419]
45Ø REM ::::::::::::::::::::::::         [1736]
46Ø MEMORY &3FAF                         [744]
47Ø DIM color(17):FOR H=Ø TO 15:C        [3816]
OLOR(H)=H:NEXT:INK Ø,Ø:INK 1,24:M
ODE 2
48Ø xr$=CHR$(23)+CHR$(1):nr$=CHR$        [3Ø91]
(23)+CHR$(Ø)
49Ø tr$=CHR$(22)+CHR$(1):no$=CHR$        [1638]
(22)+CHR$(Ø)
5ØØ REM ::::::::::::::::::::::::         [1736]
51Ø REM :                      :         [419]
52Ø REM :    QUESTIONNAIRE     :         [1361]
53Ø REM :                      :         [419]
54Ø REM ::::::::::::::::::::::::         [1736]
55Ø LOCATE 1,1:PRINT"PREMIERE CHO        [5912]
SE : EDITER LA PALETTE DE COULEUR
 UTILISEE POUR VOTRE LOGICIEL."
56Ø LOCATE 1,4:PRINT CHR$(242)+"         [4133]
et "+CHR$(243)+" POUR LE NUMERO D
E L'ENCRE."
57Ø LOCATE 1,5:PRINT CHR$(24Ø)+"         [5223]
et "+CHR$(241)+" POUR LE NUMERO D
E LA COULEUR.
58Ø LOCATE 1,6:PRINT "PUIS <ENTER        [3751]
> QUAND C'EST FINI...."
59Ø LOCATE 28,8:PRINT "APPUYEZ SU        [3283]
R UNE TOUCHE SVP":CALL &BB18
6ØØ GOSUB 122Ø                           [871]
```

```
610 PLOT 320,350:DRAW 320,16:FOR [3173]
H=0 TO AFF:LOCATE 46,H+6
620 PRINT "STYLO NR ";:PRINT USIN [5040]
G "##";H;:PRINT " = ";:PRINT USIN
G "##";COLOR(H):NEXT
630 LOCATE 1,1:PRINT"IL Y A 4 STY [9260]
LOS POSSIBLES POUR CHAQUE LETTRE.
LE CHOIX DOIT SE FAIRE EN ALLANT
DUHAUT VERS LE BAS DE LA LETTRE..
."
640 FOR H=6 TO 9 [804]
650 LOCATE 1,H:PRINT SPACE$(40):L [9870]
OCATE 3,H:PRINT "TRAIT NR "+CHR$(
24);H-5;:PRINT CHR$(24)+" AVEC LE
 STYLO NR :"
660 X1=35:Y1=H:LG=2:GOSUB 1480:EC [4308]
(H-5)=VAL(LIG$):IF EC(H-5)>AFF TH
EN PRINT CHR$(7):GOTO 650
670 NEXT H [372]
680 LOCATE 1,15:PRINT"ADRESSE DE [5892]
STOCKAGE DU FICHIER":LOCATE 1,16:
PRINT"MINIMUM &4000 : MAXIMUM &A1
E0 : &":X1=34:Y1=16:LG=4:GOSUB 14
80
690 A$=LIG$:ADD=VAL("&"+RIGHT$(A$ [3691]
,2))+VAL("&"+LEFT$(A$,2))*256
700 REM :::::::::::::::::::::::: [1736]
710 REM :                      : [419]
720 REM :  CREATION FICHIER    : [2209]
730 REM :                      : [419]
740 REM :::::::::::::::::::::::: [1736]
750 MODE MO:FOR H=0 TO AFF:INK H, [2351]
COLOR(H):NEXT
760 LOCATE 1,1:PEN 1:FOR H=200 TO [3378]
 210:PRINT CHR$(H);:NEXT
770 LOCATE 1,2:PEN 1:FOR H=211 TO [3234]
 221:PRINT CHR$(H);:NEXT
780 IF MO=0 THEN LG=400 ELSE LG=2 [1290]
00
790 FOR H=0 TO LG STEP 2:FOR G=1 [2194]
TO 4
800 IF TEST(H,401-(G*4))=1 THEN P [1967]
LOT H,401-(G*4),EC(G)
810 IF TEST(H,403-(G*4))=1 THEN P [4322]
LOT H,403-(G*4),EC(G)
820 IF TEST(H,385-(G*4))=1 THEN P [2998]
LOT H,385-(G*4),EC(G)
830 IF TEST(H,387-(G*4))=1 THEN P [3218]
LOT H,387-(G*4),EC(G)
840 NEXT G,H [389]
850 DATA dd,5e,00,dd,56,01,dd,6e, [3690]
02,dd,66,03,06,08,C5,E5,06,02,7E,
12,13,23,10,FA,E1,CD,26,BC,C1,10,
EF,C9
860 DATA dd,5e,00,dd,56,01,dd,6e, [3583]
02,dd,66,03,06,08,C5,E5,06,02,1A,
77,23,13,10,FA,E1,CD,26,BC,C1,10,
EF,C9
870 RESTORE 850:FOR H=&3FB0 TO &3 [2101]
FB0+31:READ A$:POKE H,VAL("&"+A$)
:NEXT
880 RESTORE 860:FOR H=ADD TO ADD+ [4068]
31:READ A$:POKE H,VAL("&"+A$):NEX
T
890 DEP=ADD+32:dep1=dep [1544]
900 ecr=&C000:FOR h=1 TO 22:CALL [3113]
&3FB0,ECR,DEP:ECR=ECR+2:DEP=DEP+1
6:NEXT
910 ecr=&C050:FOR h=1 TO 22:CALL [4181]
&3FB0,ECR,DEP:ECR=ECR+2:DEP=DEP+1
6:NEXT
920 DEF FN PO(X,Y)=&C000+(Y-1)*80 [585]
+(X-1)*2
930 DATA 2,10,"C>EST TERMINE IL N [3357]
E RESTE PLUS QU>A "
940 DATA 5,12,"SAUVEGARDER LE FIC [2333]
HIER <<<<<"
950 DATA 7,14,"ENTER  POUR LA SAU [2062]
VEGARDE"
960 DATA 7,16,"OU SPACE POUR RECO [1569]
MMENCER"
970 CLS:RESTORE 920:FOR H=1 TO 4: [1492]
READ ZX,ZY,A$
980 A$=UPPER$(A$):FOR T=1 TO LEN( [2706]
A$):SP=(ASC(MID$(A$,T,1)))-48
990 IF sp<0 OR sp>42 THEN sp=43 [1232]
1000 CALL ADD,FN PO(ZX,ZY)+(T*2), [2380]
DEP1+(SP*16):NEXT T,H
1010 A$=INKEY$:IF A$="" THEN 1010 [1470]
1020 IF A$=CHR$(13) THEN 1090 [730]
1030 IF A$=" " THEN RUN 470 ELSE [1254]
1010
1040 REM :::::::::::::::::::::::: [1736]
1050 REM :                      : [419]
1060 REM : SAUVEGARDE FICHIER   : [1448]
1070 REM :                      : [419]
1080 REM :::::::::::::::::::::::: [1736]
1090 MODE 2:INK 1,24:PEN 1 [1969]
1100 LOCATE 1,1:PRINT "DEBUT DE F [3644]
ICHIER EN   : &";:PRINT HEX$(ADD)
1110 LOCATE 1,2:PRINT "LONGUEUR T [3037]
OTALE DU FICHIER : 720 OCTETS"
1120 LOCATE 1,3:PRINT "STYLOS UTI [3798]
LISES...";:FOR H=1 TO 4:PRINT EC(
H);:NEXT
1130 LOCATE 1,5:PRINT "NOM DE SAU [2883]
VEGARDE : ":X1=21:Y1=5:LG=12:GOSU
B 1480
1140 LOCATE 1,7:PRINT "UNE TOUCHE [3703]
  S V P....":CALL &BB18
1150 SAVE "!"+LIG$,B,ADD,720 [2002]
1160 LOCATE 1,8:PRINT "SAUVEGARDE [3549]
 TERMINEE...":CALL &BB18:END
1170 REM :::::::::::::::::::::::: [1736]
1180 REM :                      : [419]
1190 REM :  RECHERCHE COULEUR   : [1718]
1200 REM :                      : [419]
1210 REM :::::::::::::::::::::::: [1736]
1220 MODE 0:eff=20:maxi=20:aff=15 [2618]
1230 PEN 0:FOR h=1 TO eff:LOCATE [2908]
h,1:PRINT CHR$(143):NEXT
1240 FOR h=0 TO aff:INK H,COLOR(H [7143]
):PEN H:LOCATE 5+h,1:PRINT CHR$(2
55):NEXT:PEN 1:LOCATE 1,1:PRINT "
0000"
1250 PEN 1:LOCATE 5,1:PRINT tr$;C [2463]
HR$(254):pex=5:pez=5
1260 a$=INKEY$:IF a$="" THEN 1260 [1397]
1270 IF a$=CHR$(13) THEN 1410 [730]
1280 IF a$=CHR$(243) AND pex<maxi [3193]
 THEN pex=pex+1:GOTO 1330
1290 IF a$=CHR$(242) AND pex>5 TH [2225]
```

```
EN pex=pex-1:GOTO 1330
1300 IF a$=CHR$(240) AND color(pe [2008]
x-5)<26 THEN 1380
1310 IF a$=CHR$(241) AND color(pe [879]
x-5)>0 THEN 1400
1320 PRINT CHR$(7):GOTO 1260 [1604]
1330 PEN 0:LOCATE pez,1:PRINT CHR [3500]
$(254):PEN 1:LOCATE pex,1:PRINT C
HR$(254)
1340 PEN 1:pez=pex:LOCATE 1,1:PRI [3955]
NT no$:col=color(pex-5)
1350 IF pex-5<10 THEN LOCATE 1,1: [5182]
PRINT "0";:PRINT USING "#";pex-5
ELSE LOCATE 1,1:PRINT USING "##";
pex-5
1360 IF col<10 THEN LOCATE 3,1:PR [4970]
INT "0";:PRINT USING "#";col ELSE
 LOCATE 3,1:PRINT USING "##";col
1370 LOCATE 1,1:PRINT tr$:GOTO 12 [1407]
60
1380 col=color(pex-5):col=col+1:c [3084]
olor(pex-5)=col:INK pex-5,col
1390 PEN pex-5:LOCATE pex,1:PRINT [2383]
 CHR$(255):GOTO 1340
1400 col=color(pex-5):col=col-1:c [3000]
olor(pex-5)=col:INK pex-5,col:GOT
O 1390
1410 PEN 0:FOR h=1 TO eff:LOCATE [2908]
h,1:PRINT CHR$(143):NEXT
1420 INK 0,0:INK 1,24:MODE 2:PEN [904]
1:RETURN
1430 REM ::::::::::::::::::::: [1736]
1440 REM :                   : [419]
1450 REM :   COMMANDE INPUT  : [497]
1460 REM :                   : [419]
1470 REM ::::::::::::::::::::: [1736]
1480 lig$="":LOCATE x1,y1:zr=134 [1664]
1490 a$=INKEY$:zr=zr XOR 15:PRINT [3998]
 CHR$(zr);CHR$(8);: IF a$="" THEN
 1490
1500 a$=UPPER$(a$):IF ASC(a$)=13 [9882]
THEN 1520 ELSE IF (a$>"Z" AND ASC
(a$)<>127) THEN 1490 ELSE IF ASC(
a$)=127 AND LEN (lig$)>0 THEN lig
$=LEFT$(lig$,LEN(lig$)-1):PRINT C
HR$(8);"  ";CHR$(8);CHR$(8);:GOTO
 1490
1510 IF LEN(lig$)=lg THEN PRINT C [5963]
HR$(7);:GOTO 1490 ELSE lig$=lig$+
a$:PRINT RIGHT$(lig$,1);:GOTO 149
0
1520 LOCATE x1,y1:PRINT CHR$(24)+ [3785]
" "+lig$+" "+CHR$(24)
1530 RETURN [555]
```



VERIFICATEURS V.2

Le saviez-vous? La présence dans nos listings de numéros entre crochets, assuré la détection d'éventuelles erreurs lors de la saisie d'un programme Basic.
Tapez minutieusement et sauvegardez le listing vérificateur relatif à votre machine (464, 664 ou 6128), conservez-le précieusement et lancez-le avant toute frappe ou vérification de listing; les indications nécessaires sont incluses.
Chaque somme affichée devra rigoureusement correspondre à celle figurant dans nos colonnes.

Nota : les deux traits verticaux présents en ligne 80 et 90, correspondent à une barre verticale sur un clavier QWERTY, ou à un "ù" sur clavier AZERTY.
Ces caractères - CHR$(124) - s'obtiennent par appui simultané sur SHIFT/traits verticaux.

```
10 REM Verificateur 464
20 MEMORY &A4FF
30 FOR a%=&A500 TO &A607
40 READ byte$
50 POKE a%,VAL("&"+byte$)
60 NEXT
70 CLS:LOCATE 5,1:PRINT CHR$(24)
" Verificateur V.2 464 installe
"CHR$(24):PRINT:PRINT"Chaque val
idation affiche desormais une so
mme de controle.":PRINT
80 PRINT"|CHECK,2 : liste le pro
gramme avec les            somm
es de controle.":PRINT:PRINT"|CH
ECK,8 : imprime le programme ave
c les            sommes de contro
le.":PRINT:PRINT"    "CHR$(24)"
Taper ou charger un programme. "
CHR$(24)
90 CALL &A500:|ON
100 NEW
110 DATA 21,09,a5,01,0d,a5,c3,d1
120 DATA bc,00,00,00,00,18,a5,c3
130 DATA 2a,a5,c3,2f,a5,c3,43,a5
140 DATA 4f,ce,4f,46,c6,43,48,45
150 DATA 43,cb,00,00,cf,98,aa,c3
160 DATA a8,a5,21,27,a5,18,03,21
170 DATA 24,a5,28,06,cd,00,b9,c3
180 DATA 06,dd,11,3a,bd,01,03,00
190 DATA ed,b0,c9,4f,cd,00,b9,0d
200 DATA 28,08,0d,20,ea,dd,7e,02
210 DATA 18,04,7b,11,01,00,cd,a2
220 DATA c1,cd,a3,e7,e5,4e,23,46
230 DATA 23,5e,23,56,e1,78,b1,c8
240 DATA cd,3c,c4,e5,09,e3,cd,63
250 DATA e1,21,a4,ac,cd,7a,a5,e1
260 DATA 18,e2,e5,cd,ba,a5,e3,cd
270 DATA 96,a5,cd,96,f2,e3,cd,f6
280 DATA a5,cd,4e,c3,e1,7e,a7,c8
290 DATA cd,98,a5,cd,4e,c3,18,f5
300 DATA 3a,24,ac,d6,08,47,7e,a7
310 DATA c8,cd,45,e1,23,10,f7,c9
320 DATA cd,24,a5,f5,c5,d5,e5,cd
330 DATA ba,a5,cd,f6,a5,e1,d1,c1
340 DATA f1,c9,eb,1b,af,47,67,6f
350 DATA 2f,32,23,a5,13,1a,d6,30
360 DATA 38,04,fe,0a,38,f6,1a,13
370 DATA a7,c8,4f,3a,23,a5,a1,fe
380 DATA 20,28,f3,79,fe,22,20,07
390 DATA 3a,23,a5,2f,32,23,a5,3a
400 DATA 23,a5,a7,79,c4,8a,ff,4f
410 DATA ad,07,6f,09,18,d8,3e,20
420 DATA cd,5c,c3,3e,5b,cd,5c,c3
430 DATA cd,79,ee,3e,5d,c3,5c,c3
```

```
10 REM Verificateur 664
20 MEMORY &A4FF
30 FOR a%=&A500 TO &A607
40 READ byte$
50 POKE a%,VAL("&"+byte$)
60 NEXT
70 CLS:LOCATE 5,1:PRINT CHR$(24)
" Verificateur V.2 664 installe
"CHR$(24):PRINT:PRINT"Chaque val
idation affiche desormais une so
mme de controle.":PRINT
80 PRINT"|CHECK,2 : liste le pro
gramme avec les            somm
es de controle.":PRINT:PRINT"|CH
ECK,8 : imprime le programme ave
c les            sommes de contro
le.":PRINT:PRINT"    "CHR$(24)"
Taper ou charger un programme. "
CHR$(24)
90 CALL &A500:|ON
100 NEW
110 DATA 21,09,a5,01,0d,a5,c3,d1
120 DATA bc,00,00,00,00,18,a5,c3
130 DATA 2a,a5,c3,2f,a5,c3,43,a5
140 DATA 4f,ce,4f,46,c6,43,48,45
150 DATA 43,cb,00,00,cf,02,ac,c3
160 DATA a8,a5,21,27,a5,18,03,21
170 DATA 24,a5,28,06,cd,00,b9,c3
180 DATA 4c,cb,11,5b,bd,01,03,00
190 DATA ed,b0,c9,4f,cd,00,b9,0d
200 DATA 28,08,0d,20,ea,dd,7e,02
210 DATA 18,04,7b,11,01,00,cd,a9
220 DATA c1,cd,69,e8,e5,4e,23,46
230 DATA 23,5e,23,56,e1,78,b1,c8
240 DATA cd,75,c4,e5,09,e3,cd,59
250 DATA e2,21,8a,ac,cd,7a,a5,e1
260 DATA 18,e2,e5,cd,ba,a5,e3,cd
270 DATA 98,a5,cd,58,f3,e3,cd,f6
280 DATA a5,cd,9b,c3,e1,7e,a7,c8
290 DATA cd,98,a5,cd,9b,c3,18,f5
300 DATA 3a,09,ac,d6,08,47,7e,a7
310 DATA c8,cd,22,e2,23,10,f7,c9
320 DATA cd,24,a5,f5,c5,d5,e5,cd
330 DATA ba,a5,cd,f6,a5,e1,d1,c1
340 DATA f1,c9,eb,1b,af,47,67,6f
350 DATA 2f,32,23,a5,13,1a,d6,30
360 DATA 38,04,fe,0a,38,f6,1a,13
370 DATA a7,c8,4f,3a,23,a5,a1,fe
380 DATA 20,28,f3,79,fe,22,20,07
390 DATA 3a,23,a5,2f,32,23,a5,3a
400 DATA 23,a5,a7,79,c4,ab,ff,4f
410 DATA ad,07,6f,09,18,d8,3e,20
420 DATA cd,a3,c3,3e,5b,cd,a3,c3
430 DATA cd,49,ef,3e,5d,c3,a3,c3
```

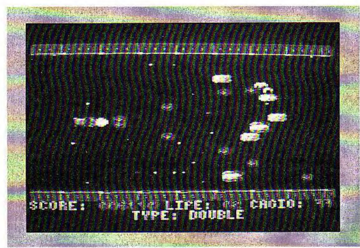
```
10 REM Verificateur 6128
20 MEMORY &A4FF
30 FOR a%=&A500 TO &A607
40 READ byte$
50 POKE a%,VAL("&"+byte$)
60 NEXT
70 CLS:LOCATE 4,1:PRINT CHR$(24)
" Verificateur V.2 6128 installe
"CHR$(24):PRINT:PRINT"Chaque va
lidation affiche desormais une s
omme de controle.":PRINT
80 PRINT"|CHECK,2 : liste le pro
gramme avec les             somm
es de controle.":PRINT:PRINT"|CH
ECK,8 : imprime le programme ave
c les            sommes de contro
le.":PRINT:PRINT"    "CHR$(24)" T
aper ou charger un programme. "C
HR$(24)
90 CALL &A500:|ON
100 NEW
110 DATA 21,09,a5,01,0d,a5,c3,d1
120 DATA bc,00,00,00,00,18,a5,c3
130 DATA 2a,a5,c3,2f,a5,c3,43,a5
140 DATA 4f,ce,4f,46,c6,43,48,45
150 DATA 43,cb,00,00,cf,02,ac,c3
160 DATA a8,a5,21,27,a5,18,03,21
170 DATA 24,a5,28,06,cd,00,b9,c3
180 DATA 49,cb,11,5e,bd,01,03,00
190 DATA ed,b0,c9,4f,cd,00,b9,0d
200 DATA 28,08,0d,20,ea,dd,7e,02
210 DATA 18,04,7b,11,01,00,cd,a6
220 DATA c1,cd,64,e8,e5,4e,23,46
230 DATA 23,5e,23,56,e1,78,b1,c8
240 DATA cd,72,c4,e5,09,e3,cd,54
250 DATA e2,21,8a,ac,cd,7a,a5,e1
260 DATA 18,e2,e5,cd,ba,a5,e3,cd
270 DATA 98,a5,cd,53,f3,e3,cd,f6
280 DATA a5,cd,98,c3,e1,7e,a7,c8
290 DATA cd,98,a5,cd,98,c3,18,f5
300 DATA 3a,09,ac,d6,08,47,7e,a7
310 DATA c8,cd,1d,e2,23,10,f7,c9
320 DATA cd,24,a5,f5,c5,d5,e5,cd
330 DATA ba,a5,cd,f6,a5,e1,d1,c1
340 DATA f1,c9,eb,1b,af,47,67,6f
350 DATA 2f,32,23,a5,13,1a,d6,30
360 DATA 38,04,fe,0a,38,f6,1a,13
370 DATA a7,c8,4f,3a,23,a5,a1,fe
380 DATA 20,28,f3,79,fe,22,20,07
390 DATA 3a,23,a5,2f,32,23,a5,3a
400 DATA 23,a5,a7,79,c4,ab,ff,4f
410 DATA ad,07,6f,09,18,d8,3e,20
420 DATA cd,a0,c3,3e,5b,cd,a0,c3
430 DATA cd,44,ef,3e,5d,c3,a0,c3
```

Vive la chasse !

DEATH'S TICKET

Aux commandes du redoutable «Monster vice», Kevin Juck se couvrit de gloire en affrontant les hordes impérialistes de la planète Zégon.

Les munitions, en nombre limité, devront être utilisées à bon escient. Le valeureux pilote pourra néanmoins se réapprovisionner en cours de jeu auprès de quelques rares containers. De plus, il rencontrera aussi peu fréquemment des types d'affûts capables de renforcer l'efficité de son tir (double, triple, etc.). Les commandes sont classiques (Joystick ou touches directionnelles + espace) et «ESC» permet l'abandon. Malgré la simplicité du scénario, ce *«shoot'em up»* est remarquablement animé.



Sauvegarde

Sauvez sous un nom de votre choix, le court listing Basic de chargement. Entrez ensuite par *Amsaisie V.2* en vous reportant à son mode d'emploi, les quatre listings de codes hexadécimaux.

Nom	Adr. déb.	Long
DEATH1	&4000	&401
DEATH2	&4800	&2800
DEATH3	&8000	&ED4
DEATH4	&94E0	&238

La longueur est ici précisée à l'attention de ceux qui envisagent raisonnablement de morceler leur travail en plusieurs fichiers qui devront ultérieurement être réunis en quatre fichiers définitifs.

Sébastien Royer

```
1 ' ******************** [665]
2 ' ** DEATH'S TICKET ** [700]
3 ' ** (C) R.O.S.SOFT ** [2034]
4 ' *        by        * [227]
5 ' * Sebastien  Royer * [1874]
6 ' ******************** [665]
7 ' Special thanks to:   [838]
8 ' Robert,Vincent,Steph [1509]
ane
9 ' Nelly,Aude,Stephanie [1848]
,Severine
10 ' and MARIE           [384]
11 OPENOUT"MARIE":MEMORY [3263]
 &FFF:CLOSEOUT
12 LOAD"DEATH1",&1000    [1394]
13 LOAD"DEATH2",&1800    [1356]
14 LOAD"DEATH3",&8000    [1092]
15 LOAD"DEATH4",&94E0    [1495]
16 CALL &8000            [453]
```

```
4000:00 00 00 00 00 00 00 00:40
4008:00 00 00 00 00 00 00 00:48
4010:04 00 04 00 44 00 44 00:90
4018:10 00 00 00 10 00 00 00:B8
4020:08 08 88 20 00 00 00 00:90
4028:00 00 00 00 00 00 00 00:68
4030:04 00 04 00 4C 88 44 00:08
4038:CC 20 10 00 10 00 00 00:04
4040:0C 88 4C 00 4C 00 CC 88:D0
4048:44 88 44 20 30 20 00 00:D8
4050:08 08 08 88 00 88 44 00:EC
4058:10 00 20 00 20 88 20 20:B8
4060:0C 00 08 00 4C 88 88 88:68
4068:CC 20 00 20 10 20 00 00:44
4070:00 08 00 88 10 00 00 00:30
4078:00 00 00 00 00 00 00 00:B8
4080:00 08 04 00 04 00 44 00:CC
4088:44 00 10 00 00 20 00 00:FC
4090:04 00 00 08 00 08 00 88:64
4098:00 88 00 20 10 00 00 00:B8
40A0:00 00 00 00 00 00 08 44:38
40A8:04 88 44 88 20 10 00 00:E8
40B0:00 00 00 00 04 00 04 00:20
40B8:4C 88 10 00 10 00 00 00:D4
40C0:00 00 00 00 00 00 00 00:00
40C8:00 00 00 08 00 88 10 00:C8
40D0:00 00 00 00 00 00 00 00:10
40D8:4C 20 00 00 00 00 00 00:A4
40E0:00 00 00 00 00 00 00 00:20
40E8:00 00 04 00 10 00 00 00:84
40F0:00 08 00 08 00 88 44 00:6C
40F8:44 00 88 00 20 00 20 00:94
4100:41 00 82 02 82 02 02 A8:F4
4108:02 A8 A8 A8 54 00 00 00:D7
4110:41 00 C3 00 01 00 01 00:E7
4118:54 00 54 00 54 00 00 00:4D
4120:41 00 82 82 00 02 01 00:43
4128:02 00 A8 00 FC A8 00 00:3F
4130:41 00 82 02 00 02 01 00:53
4138:00 02 A8 A8 54 00 00 00:B9
4140:82 00 82 02 82 02 03 02:4C
4148:00 A8 00 A8 00 A8 00 00:69
4150:C3 82 82 00 02 00 01 00:EF
4158:00 02 A8 A8 54 00 00 00:D9
4160:41 00 82 02 82 00 03 00:0F
4168:02 A8 A8 A8 54 00 00 00:37
4170:C3 82 00 82 00 02 01 02:A3
4178:00 A8 00 A8 00 A8 00 00:99
4180:41 00 82 82 82 02 01 00:2D
4188:02 A8 02 A8 54 00 00 00:65
4190:41 00 82 82 82 02 01 02:4D
4198:00 A8 A8 A8 54 00 00 00:65
41A0:00 00 00 00 04 00 44 00:D1
41A8:00 00 04 00 10 00 00 00:45
41B0:00 00 00 00 04 00 44 00:E1
41B8:00 00 04 00 10 00 20 00:35
41C0:00 00 00 08 04 00 08 00:6D
41C8:88 00 44 00 00 20 00 00:1D
41D0:00 00 00 00 00 00 00 00:11
41D8:0C 88 00 00 98 20 00 00:ED
41E0:00 00 08 00 04 00 00 08:8D
41E8:00 88 44 00 20 00 00 00:A5
41F0:04 00 08 08 00 88 44 00:79
41F8:10 00 00 00 10 00 00 00:99
4200:04 00 4C 08 4C 88 88 88:EE
4208:20 20 20 00 10 20 00 00:1A
4210:04 00 08 08 08 88 CC 88:BA
4218:88 20 88 20 20 20 00 00:9A
4220:0C 00 08 08 08 88 CC 00:92
```

```
4228:88 20 88 20 30 00 00 00:3A
4230:04 00 08 08 08 00 88 00:8E
4238:88 00 20 20 10 00 00 00:32
4240:0C 00 08 88 08 88 88 20:D6
4248:88 20 88 20 30 00 00 00:5A
4250:0C 08 08 00 08 00 CC 00:82
4258:88 00 20 00 30 20 00 00:32
4260:04 00 08 08 08 00 CC 00:9A
4268:88 00 20 00 20 00 00 00:32
4270:04 00 08 08 08 00 88 88:0E
4278:88 20 20 20 10 00 00 00:B2
4280:08 08 08 88 08 88 CC 88:3E
4288:88 20 88 20 20 20 00 00:0A
4290:0C 08 04 00 44 00 44 00:2A
4298:44 00 10 00 30 20 00 00:FE
42A0:04 08 00 08 00 08 00 88:86
42A8:00 88 20 20 10 00 00 00:2A
42B0:08 08 08 88 08 88 CC 00:2E
42B8:88 88 88 20 20 20 00 00:0A
42C0:08 00 08 00 88 00 88 00:82
42C8:88 00 20 00 30 20 00 00:A2
42D0:08 08 4C 88 4C 88 88 88:D2
42D8:88 20 20 20 20 20 00 00:22
42E0:08 08 08 88 4C 88 CC 88:F2
42E8:CC 20 88 20 20 20 00 00:AE
42F0:04 00 08 08 08 88 88 88:BE
42F8:88 20 20 20 10 00 00 00:32
4300:0C 00 08 08 08 88 CC 00:73
4308:88 00 20 00 20 00 00 00:D3
4310:04 00 08 08 08 88 88 88:DF
4318:88 88 20 20 10 88 00 00:53
4320:0C 00 08 08 08 88 CC 00:93
4328:88 88 88 20 20 20 00 00:7B
4330:04 00 08 08 08 00 44 00:B3
4338:00 88 20 20 10 00 00 00:BB
4340:0C 08 04 00 44 00 44 00:DB
4348:44 00 10 00 10 00 00 00:4F
4350:08 08 08 88 88 88 88 88:B3
4358:88 20 20 20 10 00 00 00:93
4360:08 08 08 88 88 88 88 88:C3
4368:20 20 10 00 10 00 00 00:8B
4370:08 08 08 08 08 88 88 88:53
4378:CC 20 98 20 20 20 00 00:6F
4380:08 08 08 08 08 88 44 00:47
4388:88 88 88 20 20 20 00 00:DB
4390:08 08 08 88 88 88 44 00:D7
4398:44 00 10 00 10 00 00 00:9F
43A0:0C 08 00 08 00 88 44 00:2B
43A8:88 00 20 00 30 20 00 00:83
43B0:04 08 04 00 04 00 44 00:03
43B8:44 00 10 00 10 20 00 00:7F
43C0:08 00 08 00 88 00 44 00:A7
43C8:44 00 00 88 00 20 00 20:2F
43D0:04 08 00 08 00 88 00 88:B7
43D8:00 88 00 20 10 20 00 00:BB
43E0:00 00 00 08 04 00 0C CC:0B
43E8:CC 98 10 00 00 20 00 00:17
43F0:64 0C 44 04 00 00 64 8C:A7
43F8:00 00 64 0C 44 88 30 8C:CB
4400:84 00 00 00 00 00 00 00:C8
```

```
4800:EC 8B F3 3B 20 CD C3 35:7A
4808:73 AC 1D 15 1A 9B FA 24:E0
4810:E7 95 A2 23 75 B2 59 23:D7
4818:E1 DF C2 3D BC 63 3A 11:55
4820:D0 2E FC 63 73 75 A4 42:9D
4828:10 46 08 D7 11 EB 9E E6:D9
4830:24 46 D4 03 1D B4 CC 5F:05
4838:F6 0B 6C B2 16 61 CA C6:02
4840:0F 73 55 49 2F CC 9E 9C:85
4848:93 D8 67 14 7E 43 C2 5E:9E
4850:94 16 E2 BE 15 51 B0 AF:8D
4858:B1 76 89 8A 84 F6 62 C6:36
4860:D0 48 AE 94 A7 E5 C4 94:FF
4868:3F 2B 9D F6 73 57 2D 8D:E0
4870:84 52 85 20 C0 65 3B BC:8A
4878:17 FB D1 33 2F AB D2 E5:DF
4880:4A C2 64 51 02 23 D5 7C:95
4888:B1 9D A1 93 78 DA 05 91:09
4890:D2 55 51 D7 0B BB 16 26:06
4898:61 70 39 4C 6B B4 28 4F:DB
48A0:A3 58 E6 74 FC 9E F8 71:AD
48A8:44 62 5C 19 11 F3 D8 40:5F
48B0:EA 90 56 40 5B 68 FD 55:CE
48B8:A1 88 BC B1 44 49 A8 7F:43
48C0:6D E8 8A A5 70 F2 8B 27:58
48C8:F9 D8 4F 06 55 A6 BD 4B:CE
48D0:45 AE A9 D1 A1 28 B7 99:D6
48D8:AC 62 02 2F 51 AD 79 23:5C
48E0:2F 66 B6 43 D1 EE 1D 35:6D
48E8:F1 0D CB 52 3E 9C A7 A8:93
48F0:1E 88 8B 25 D6 D7 08 44:2B
48F8:77 87 97 4E B5 81 C7 C6:F2
4900:4A AD A6 24 F3 C7 EF 1B:39
4908:93 69 6C 2D C8 62 A8 8F:F2
4910:D7 0D E9 C5 6F 73 25 43:11
4918:7D BC D1 EB C6 EB 79 D1:AC
4920:E9 AB 1C 29 6C 72 FC AD:B4
4928:64 B4 D5 22 4A 18 4E 7D:50
4930:2C 40 14 3A A1 66 76 F8:CC
4938:5A DF 0D 4F 38 DC BF A0:75
4940:40 4F 00 95 00 CC D4 12:DF
4948:CE 2E 6B 21 77 AE 33 C3:64
4950:12 A9 20 5E 3A 0E 12 3D:B1
4958:0D 8B A5 B5 94 86 3F B2:D8
4960:EA 69 FE DF 4F A0 55 0C:D9
4968:A3 C4 B2 DB 05 80 3A 84:2D
4970:B5 87 5F 85 EE 89 79 67:10
4978:3B 8C C3 11 19 1B 24 45:E4
4980:82 3D A6 BE 09 6A 15 CE:5B
4988:A7 14 36 4E 31 34 C8 A3:37
4990:A6 31 39 1C D9 E8 61 5B:28
4998:9C BB 9C A8 CB F0 58 E3:7E
49A0:76 CF FC 5C 3B 0E 79 28:6B
49A8:A7 90 2B F7 04 85 9C 21:93
49B0:9D C7 04 18 15 64 CA 58:97
49B8:51 03 D6 47 C9 2D E9 69:98
49C0:5F C5 36 F1 41 87 E7 70:98
49C8:58 C1 53 39 19 11 AD A3:7E
49D0:8B 6E 2F 67 8B 6E C5 C5:7F
49D8:A1 EB F4 F0 4E B9 69 B7:A7
49E0:02 71 F2 6B 79 64 80 B0:44
49E8:99 A1 D9 06 7D 18 61 F6:07
49F0:04 AD A5 1F BE 89 F7 0B:07
49F8:D6 AC F7 EA 44 43 30 18:F2
4A00:D3 D4 3B 3A 46 B2 A8 FA:50
4A08:A5 AD F5 67 2A 63 DF 91:91
4A10:EB E8 D1 0A D1 5A FD 28:0C
4A18:D4 AF 7C 6F D0 09 E3 B7:F7
4A20:FC E9 27 01 EA E1 7D 64:14
4A28:C7 4E EA 07 DA 41 C6 4F:59
4A30:1F 77 2E 58 62 FE 46 F5:E1
4A38:6C E8 0B 3D E1 4B A5 4E:ED
4A40:2B 25 0F B7 CE BD 94 8E:F8
4A48:0C 74 97 F6 A4 64 33 DC:F4
4A50:58 69 A4 F3 24 55 8C AF:7A
4A58:08 B2 2E 2B 15 8D 28 A9:5B
4A60:27 2C 91 CD 94 AD 6D 0B:55
4A68:45 E5 70 AC EB 6B 8E 02:CC
4A70:8F D7 05 B3 6F 22 DD 1E:C4
4A78:DA B1 B7 68 34 F8 82 6B:7D
4A80:2F 8B 9E AE 0E 49 A8 01:3D
4A88:D2 85 E1 36 9E FB 76 0F:D3
4A90:96 3C CD 72 2F 75 76 24:1A
4A98:BB B3 77 78 0C 0A ED 31:C3
4AA0:66 82 CF 4A 80 1B 91 93:9A
4AA8:FB FE 8D FB 37 14 B5 98:BA
4AB0:6F 7A 12 91 12 02 A1 F1:2C
4AB8:8F 79 BD D7 DB E7 4B 1D:BC
4AC0:51 38 DB C7 93 DD 7A 11:63
4AC8:94 B9 F2 11 C6 D7 EE C8:DC
4AD0:09 02 D7 B8 FD 3B A2 A4:6D
4AD8:A9 0F 77 EE 3C 07 13 9D:C9
4AE0:15 AF FD 43 F7 D3 02 67:AB
4AE8:FD CE A7 E3 71 3D 73 8A:64
4AF0:12 B0 08 52 48 44 30 53:F4
4AF8:8D C3 A8 35 B9 FA 2B F9:8F
```

```
4B00:27 1A 33 96 D6 B2 DE 70:83
4B08:74 BC F2 D6 8C 1A CC FC:39
4B10:DB 60 AC 66 C4 7E 33 93:57
4B18:EC 91 92 BB 06 FE 0C AF:F1
4B20:39 41 74 E7 27 12 DA 42:53
4B28:45 08 5D CA 0E CF DE 32:C9
4B30:37 31 D9 69 A7 B2 30 B3:9A
4B38:A9 3A 6C 11 5A 20 D8 2C:F2
4B40:4E 1E 64 65 E5 B1 B9 D8:43
4B48:00 C2 FD 6C E9 DE CF 87:60
4B50:48 E0 EB 96 1E 16 EC 88:8A
4B58:0A 37 1A 0A CC 93 35 64:92
4B60:CC 15 B3 36 29 75 7A D1:FB
4B68:69 14 3F F5 26 04 83 9F:38
4B70:35 68 64 B4 F4 6A 81 D4:23
4B78:7A 6B D1 FD B9 DC 30 2F:07
4B80:E1 FD 44 84 CC E5 AD 85:BF
4B88:19 15 10 9F 40 FB 1A 58:5A
4B90:1D 68 05 E8 76 E8 9E 10:07
4B98:78 32 C9 CA B3 50 F9 80:70
4BA0:C5 64 55 1C 61 F2 5F 5A:E1
4BA8:20 98 A1 4C 0E 44 5F 26:FD
4BB0:55 0B 71 92 97 B6 8B AF:7D
4BB8:CB C9 1F 18 06 09 B6 E8:AB
4BC0:B7 A3 10 08 C4 03 80 26:EE
4BC8:C9 C2 2D 8B 28 D8 92 A5:11
4BD0:49 BA BF 4B 59 65 00 17:14
4BD8:7E 1D F5 17 52 9F 13 5E:DF
4BE0:C1 E7 17 6B 5A 1C 6A 26:2B
4BE8:0C 36 6C 51 EF 49 91 09:D3
4BF0:81 F9 CA 28 5F 10 77 A8:68
4BF8:7E 70 FC 55 D5 6F 83 3D:29
4C00:11 4C A5 EC F0 48 CD 2D:F7
4C08:1D 49 27 49 50 84 BC 55:10
4C10:B6 D9 2A F2 D0 0A 20 2F:AE
4C18:C8 83 D5 93 5B 97 0C 04:C2
4C20:9F 72 84 33 CE CC 6B B9:CA
4C28:A8 36 2B A8 02 61 37 CC:DA
4C30:E2 CA 5C D2 78 0B 93 6A:3D
4C38:49 5F CC AD E8 E0 D4 A2:47
4C40:17 1A 45 64 9A E3 66 14:F4
4C48:2D C8 1E 9F 68 1C 58 4A:8F
4C50:5C BC D3 DA EB AE 7E 58:2E
4C58:30 CC 8C 8E 69 32 D1 50:B8
4C60:53 7C 76 64 6E 82 92 3D:01
4C68:BD 81 0D D4 19 95 DB 3A:B2
4C70:E0 02 7B D8 BE 26 B9 1E:0A
4C78:31 EE C8 E2 F1 D6 8E 0F:C4
4C80:F5 9E 9F 11 9D 7A 9D 87:8E
4C88:05 86 BB 88 AE 2F A0 18:D6
4C90:F1 20 13 46 D0 91 EF DE:4D
4C98:0B 0E 43 80 A0 9C E9 81:03
4CA0:72 9C 6E 44 C3 3C C6 1A:61
4CA8:2E 00 D2 93 4D 79 31 BA:62
4CB0:63 0E D3 C6 21 71 58 B2:4F
4CB8:E7 F5 86 52 BF 38 9B 96:A7
4CC0:A3 79 1E BD E1 13 75 9B:D1
4CC8:98 D1 29 FA 8A 8F 7F 11:BE
4CD0:CC EB 94 0B 49 8E 3E A4:39
4CD8:00 4B 1C 24 87 95 3B 55:04
4CE0:FC 3E F7 0C FB F4 37 ED:41
4CE8:30 CF 81 36 81 05 58 F8:28
4CF0:16 04 36 39 A4 58 1B 5F:D9
4CF8:1E 2A 17 EA 5B 53 30 01:B4
4D00:C2 2B 90 5D 9A 2F 77 27:1E
4D08:E5 52 F1 67 05 54 DA C9:9C
4D10:E6 18 3F ED C2 66 62 AA:10
4D18:A0 65 58 92 49 E4 B5 E9:1F
4D20:B3 B0 8E 0C 4D 3C EC 05:DF
4D28:C6 C8 36 1D 1D 04 0A 3B:A8
4D30:91 15 B2 F6 42 C6 60 6D:1C
4D38:F8 D8 E6 43 6E C4 81 66:60
4D40:C8 8D EA CD 13 79 62 BA:14
4D48:B8 CA 74 55 AD 96 F4 F5:CA
4D50:4C C8 68 C3 8F 82 16 98:EE
4D58:20 74 2B EF D0 0C 2C 52:06
4D60:0F 96 F6 98 BD B1 14 F3:25
4D68:22 1C B5 DF E8 BB A8 A1:34
4D70:F2 B4 89 52 BB C3 25 72:C6
```

```
4D78:68 5C D2 B8 77 BB CF F8:59
4D8Ø:53 CØ 52 FA Ø6 2E 2F 83:11
4D88:DA 33 4F 18 B2 73 32 B7:A4
4D9Ø:2C 3B 1E E2 58 Ø4 D8 A6:49
4D98:83 5F AE 6B BØ 2C C6 6C:1E
4DAØ:F8 Ø4 C4 2C Ø3 AB 48 DØ:72
4DA8:92 2D 35 97 8Ø 74 8F A4:1D
4DBØ:19 5F 34 33 EC CØ 98 Ø1:88
4DB8:3C 9F EØ E7 45 1D 49 C6:F1
4DCØ:9B F2 6Ø 2A E7 ED 6E 7B:3F
4DC8:3F DØ AC 97 55 DB C7 D3:28
4DDØ:4D ØE 82 EB AE DB 54 6E:FC
4DD8:8B 89 6A D6 DØ B2 A4 3B:E8
4DEØ:11 7C Ø3 E4 7A 64 B7 A4:AA
4DE8:94 FD E2 3A B3 Ø1 23 BC:AB
4DFØ:8C EA 24 11 CA 98 EE FE:41
4DF8:65 C3 FA 81 77 31 8E 26:AD
4EØØ:Ø4 C7 3C 16 1E 1Ø 77 A3:3B
4EØ8:8F 4A 6C EØ 66 55 7E 13:43
4E1Ø:Ø2 FC 6F 33 25 Ø8 F7 17:D3
4E18:C5 E9 1D 57 F5 31 A4 7E:ØB
4E2Ø:85 BB CB 33 F4 B1 51 7F:AF
4E28:DA 37 Ø4 FA 11 29 1C 3B:99
4E3Ø:E5 B2 D8 6D 3Ø 7E 59 11:DE
4E38:14 8C 1E 94 8Ø 8Ø 77 4Ø:1D
4E4Ø:F9 69 96 86 2A 15 5B 9A:DØ
4E48:5E C9 F7 F1 77 4F EE Ø6:ØE
4E5Ø:Ø4 45 EØ B3 2B F1 5B 58:52
4E58:F1 4F A4 A5 52 7F 5A 5Ø:3F
4E6Ø:7B 3E EF D2 ØE 73 87 C3:7B
4E68:EF 82 E6 26 29 5D CØ A8:6E
4E7Ø:E3 55 A9 BF 87 7F AB C6:BC
4E78:C4 4E 2D E2 71 7E DA E5:7C
4E8Ø:A7 9C 43 EØ 84 5F 82 B9:1A
4E88:7C 22 EB D6 1B 57 1B 73:95
4E9Ø:7Ø Ø1 AC 9A 16 DB FD 3Ø:B7
4E98:ØF 9C FD B2 49 FB 2Ø D8:DB
4EAØ:BA 94 2F E3 48 93 Ø8 18:BB
4EA8:5B F7 D2 36 73 Ø6 E7 3F:39
4EBØ:C4 F8 B6 9D BD 19 68 69:AF
4EB8:28 69 74 63 C5 CØ EØ Ø9:A9
4ECØ:84 26 D2 14 BB E8 9A 62:Ø1
4EC8:CA 66 BC 46 FC BF BØ 8E:9E
4EDØ:FC 85 5D DC 5B 8C A6 3C:24
4ED8:52 E4 8D 33 32 EE D4 CF:85
4EEØ:E3 F3 85 13 BF 7Ø CF 1Ø:56
4EE8:E7 6C B1 84 EC 99 18 EE:62
4EFØ:EB 74 E5 7B 9B FØ BE FB:5D
4EF8:77 6Ø B2 69 EF 4F 33 33:B9
4FØØ:FE 3C 75 F3 75 Ø4 9C 59:5D
4FØ8:5E 5A 69 5Ø C8 ØC 6B 7E:F1
4F1Ø:F1 Ø9 Ø5 1C C8 47 69 18:12
4F18:DF 44 66 13 A4 34 71 EE:3F
4F2Ø:BA E1 5B 27 43 E1 D1 37:9C
4F28:3F 65 D2 45 4A C1 12 6C:EØ
4F3Ø:46 3C 9A D6 6D 1D C3 65:AF
4F38:B6 3F 2E Ø7 9C 18 A8 69:DD
4F4Ø:57 8C 48 7Ø 91 C8 8Ø E3:B3
4F48:53 AB 18 6Ø 1F 33 6B 7B:9A
4F5Ø:AC B2 4B 9A 2C 95 6C E9:8E
4F58:B7 77 C3 81 A1 ØF 37 3C:79
4F6Ø:Ø2 A5 47 BA EØ 9C Ø2 DD:B6
4F68:BC FC 58 EA 5B EB D9 2Ø:53
4F7Ø:B6 Ø9 4C F2 11 3A 78 C9:74
4F78:74 C8 47 78 F5 63 1C CE:CF
4F8Ø:98 E5 34 34 F2 F5 D7 F9:BE
4F88:4C C7 E2 63 25 A3 62 36:CC
4F9Ø:EF 23 9A 1Ø 36 22 E1 14:C3
4F98:33 8F D3 FE E9 39 5Ø 6D:24
4FAØ:8D ØC 45 CB 87 Ø3 88 CE:6C
4FA8:BE 91 7B 41 43 2C B4 A6:BF
4FBØ:D6 AB C4 17 4E DD 12 27:3D
4FB8:65 26 88 EF 94 82 49 5A:CB
4FCØ:ØE 4C 68 2A ØF D3 53 42:27
4FC8:71 65 3F E1 C6 ØD 64 97:33
4FDØ:18 FE 48 1B B4 BC C7 22:E4
4FD8:D1 Ø6 1Ø ØC 1C 42 CD 86:47
4FEØ:1F 4B 66 B3 F4 Ø3 96 CD:3A
4FE8:2F DE F2 8B 2D CE EA EE:AF
4FFØ:AB C4 59 29 77 59 98 F9:7A
4FF8:B4 1D C6 F5 52 CA 28 AØ:C9
5ØØØ:8Ø ØØ 7C 43 ØØ 3Ø ØØ C2:8Ø
5ØØ8:CØ 14 A9 92 CC 2Ø 43 C3:2C
5Ø1Ø:14 A9 C6 ØC 98 A3 Ø3 ØØ:27
5Ø18:7C Ø6 CØ ØC F1 F3 14 A9:9B
5Ø2Ø:C6 ØC 98 5Ø FØ 14 A9 92:AD
5Ø28:CC 2Ø AØ ØØ 7C 43 ØØ 3Ø:E2
5Ø3Ø:ØØ 14 BC 7C 56 29 83 7C:E5
5Ø38:56 14 BC 24 D8 48 84 8C:AE
5Ø4Ø:4C 24 D8 ØØ ØØ 4C 3Ø ØØ:C4
5Ø48:ØØ ØØ 48 52 ØØ FØ ØØ 44:78
5Ø5Ø:C1 ØØ ØØ 5Ø 2Ø 1Ø 82 4Ø:2F
5Ø58:C3 51 2Ø 7Ø Ø2 8Ø ØØ Ø2:47
5Ø6Ø:C6 2Ø ØØ 82 ØØ A2 84 FØ:A6
5Ø68:ØØ 82 FØ ØØ C2 AØ Ø2 Ø2:34
5Ø7Ø:ØØ ØØ Ø6 1Ø 82 51 43 5Ø:D7
5Ø78:88 44 C1 ØØ ØØ 5Ø 2Ø ØØ:DB
5Ø8Ø:84 52 ØØ BØ ØØ ØØ ØØ 4C:18
5Ø88:7Ø ØØ ØØ ØØ ØØ 4C 3Ø ØØ:6Ø
5Ø9Ø:ØØ ØØ 48 52 ØØ FØ ØØ 44:CØ
5Ø98:C1 ØØ ØØ 5Ø 2Ø 1Ø 82 4Ø:77
5ØAØ:83 51 2Ø 7Ø Ø2 ØØ 82 ØØ:CD
5ØA8:C6 2Ø ØØ ØØ Ø2 ØØ 84 FØ:24
5ØBØ:ØØ ØØ A2 ØØ C2 AØ Ø2 ØØ:7E
5ØB8:AØ ØØ Ø6 1Ø 82 4Ø 83 5Ø:19
5ØCØ:88 44 C1 ØØ ØØ 5Ø 2Ø ØØ:23
5ØC8:84 52 ØØ BØ ØØ ØØ ØØ 4C:6Ø
5ØDØ:7Ø ØØ ØØ AØ 2Ø 1Ø 5Ø ØØ:4Ø
5ØD8:2Ø 5Ø ØØ ØØ AØ ØØ 1Ø ØØ:78
5ØEØ:1Ø ØØ ØØ 2Ø 5Ø 2Ø 5Ø 5Ø:CØ
5ØE8:5Ø AØ 2Ø BØ ØØ AØ 2Ø 2Ø:88
5ØFØ:ØØ 2Ø AØ 5Ø 5Ø ØØ AØ AØ:9Ø
5ØF8:2Ø AØ AØ BØ BØ 18 64 3Ø:84
51ØØ:BØ 24 7Ø 64 7Ø 7Ø BØ 24:E9
51Ø8:BØ BØ 9Ø 6Ø 7Ø 24 3Ø 3Ø:71
511Ø:E4 7Ø 3Ø Ø4 88 ØØ ØØ ØØ:6D
5118:ØØ ØØ ØØ 8Ø BØ ØØ ØØ 44:F9
512Ø:88 ØØ ØØ 84 CC 44 88 DØ:D5
5128:1Ø ØØ ØØ CØ 18 DØ 1Ø 84:71
513Ø:CC ØØ ØØ CØ CC 84 98 CØ:89
5138:98 44 88 84 18 CØ CC 84:FD
514Ø:18 DØ 1Ø CØ CC 84 18 CØ:35
5148:CC 84 CC 84 98 84 CC 84:A5
515Ø:18 CØ 18 CØ CØ C1 83 CØ:5C
5158:CØ C1 83 CØ C3 83 43 CØ:2A
516Ø:C3 83 43 C1 83 56 43 C1:B7
5168:83 56 43 C1 56 A9 83 C1:F6
517Ø:56 A9 C3 C3 56 29 83 C3:ØF
5178:16 A9 83 83 A9 43 D6 83:97
518Ø:A9 43 83 83 43 C3 56 83:E8
5188:43 C3 56 43 C3 Ø3 BC 43:CD
519Ø:C3 Ø3 BC CØ C3 83 Ø3 Ø3:EC
5198:Ø3 Ø3 Ø3 D6 FC FC FC FC:EB
51AØ:FC FC BC D6 FC FC FC FC:Ø9
51A8:FC FC BC 56 A9 FC A9 FC:2D
51BØ:A9 FC BC 56 E9 FC E9 FC:E2
51B8:E9 FC BC 56 FC FC FC FC:ØE
51CØ:FC FC BC 56 FC FC FC FC:29
51C8:FC FC BC 3C 3C 3C 3C 3C:49
51DØ:3C 3C 3C ØØ 4Ø 8Ø ØØ ØØ:C9
51D8:ØØ ØØ 4Ø 4Ø 84 48 8Ø ØØ:AD
51EØ:ØØ 4Ø 84 84 4C 8C 48 8Ø:Ø9
51E8:4Ø 84 4C 4C 98 64 8C 48:F9
51FØ:84 4C 98 98 7Ø BØ 64 8C:F1
51F8:4C 98 7Ø 7Ø AØ 5Ø BØ 64:C5
52ØØ:98 7Ø AØ AØ ØØ ØØ 5Ø BØ:DA
52Ø8:7Ø AØ ØØ ØØ ØØ ØØ ØØ 5Ø:8A
521Ø:AØ ØØ ØØ ØØ 8Ø ØØ ØØ ØØ:82
5218:8Ø ØØ ØØ 4Ø 48 ØØ ØØ 4Ø:52
522Ø:48 ØØ ØØ 84 8C 8Ø ØØ 84:A6
5228:8C 8Ø ØØ 4C 64 48 4Ø 4C:FA
523Ø:64 48 4Ø 98 BØ 8C 84 98:AA
5238:BØ 8C 84 7Ø 5Ø 64 4C 7Ø:1A
524Ø:5Ø 64 4C AØ ØØ BØ 98 AØ:56
5248:ØØ BØ 98 ØØ ØØ 5Ø 7Ø ØØ:B2
525Ø:ØØ 5Ø 7Ø 4C ØØ ØØ ØØ ØØ:C2
5258:4C ØØ ØØ 98 ØØ ØØ ØØ ØØ:56
526Ø:98 ØØ ØØ ØØ ØØ ØØ 41 Ø2:21
5268:ØØ ØØ ØØ ØØ 11 8A Ø1 A8:92
527Ø:ØØ 4Ø Ø8 ØØ 45 ØA ØØ ØØ:EF
5278:67 Ø4 88 ØØ ØØ ØØ ØØ ØØ:D1
528Ø:8F ØØ ØØ 83 ØØ ØØ 84 ØØ:Ø9
5288:ØØ ØØ ØØ 56 ØØ ØØ 4C ØØ:46
529Ø:ØØ ØØ ØØ 5Ø 2Ø 5Ø 2Ø 5Ø:Ø2
5298:2Ø 5Ø 2Ø EØ 9Ø EØ 9Ø EØ:8A
52AØ:9Ø EØ 9Ø EØ CC EØ CC EØ:42
52A8:CC EØ CC 24 CC 24 CC 24:Ø2
52BØ:CC 24 CC 24 8C 24 8C 24:92
52B8:8C 24 8C 8C 8C 8C 8C 8C:EA
52CØ:8C 8C 8C CC C8 CC C8 CC:12
52C8:C8 CC C8 Ø4 8Ø Ø4 8Ø Ø4:1A
52DØ:8Ø Ø4 8Ø CØ CØ CØ CØ CØ:AA
52D8:CØ CØ CØ C3 83 C3 C3 C3:44
52EØ:C3 43 C3 Ø3 43 Ø3 Ø3 Ø3:5E
52E8:Ø3 83 Ø3 F3 F3 F3 F3 F3:C6
52FØ:F3 F3 F3 F1 F1 F1 F1 F1:32
52F8:F1 F1 F1 FØ FØ FØ FØ FØ:1Ø
53ØØ:FØ FØ FØ ØØ ØØ ØØ ØØ ØØ:F3
53Ø8:ØØ ØØ ØØ FØ FØ FØ FØ FØ:7B
531Ø:FØ FØ FØ CØ 33 ØØ ØØ CØ:Ø2
5318:C3 ØØ ØØ 91 33 ØØ ØØ C1:79
532Ø:83 ØØ ØØ 91 67 4Ø Ø8 C1:FD
5328:Ø3 4Ø Ø8 33 8F ØC 98 C3:35
533Ø:56 ØC 98 33 8F 4C 7Ø 83:4Ø
5338:56 4C 7Ø 67 ØA 1Ø AØ 83:6F
534Ø:BC 1Ø AØ 67 ØA ØØ ØØ Ø3:35
5348:A8 ØØ ØØ 27 Ø5 ØØ ØØ 56:A8
535Ø:3C ØØ ØØ 4Ø ØC CC 88 84:BB
5358:CC 98 7Ø 4C 98 3Ø FØ 44:EF
536Ø:3Ø FØ AØ 4Ø CØ CØ 8Ø CØ:63
5368:CØ E2 F3 4Ø D1 D1 A2 4Ø:81
537Ø:CØ CØ 8Ø CØ CØ E2 F3 4Ø:34
5378:D1 D1 A2 84 CC 98 7Ø 4C:3Ø
538Ø:98 3Ø FØ 44 3Ø FØ AØ CØ:9B
5388:CØ E2 F3 4Ø D1 D1 A2 ØØ:A1
539Ø:ØØ ØØ ØØ 4Ø CØ CØ 8Ø CØ:A3
5398:CØ E2 F3 C8 D1 D1 F2 4C:61
53AØ:98 3Ø FØ C4 3Ø FØ F1 4Ø:F2
53A8:F3 F3 A2 ØØ ØØ ØØ ØØ ØØ:BA
53BØ:ØØ ØØ ØØ ØØ ØØ ØØ ØØ 4Ø:Ø3
53B8:CØ CØ 8Ø CØ CØ E2 F3 C8:BC
53CØ:D1 D1 F2 44 3Ø FØ AØ ØØ:5C
53C8:ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ:1B
53DØ:ØØ ØØ ØØ 4Ø CØ D1 A2 E2:47
53D8:ØC CC D9 C8 CØ CØ DØ CØ:6A
53EØ:CØ E2 F3 4Ø D1 D1 A2 ØØ:F9
53E8:ØØ ØØ ØØ ØØ ØØ ØØ ØØ 4Ø:3B
53FØ:CØ CØ A2 CØ D1 F3 F3 4Ø:D5
53F8:ØC CC 88 84 CC 98 7Ø 4C:93
54ØØ:98 3Ø FØ 4Ø CØ CØ 8Ø CØ:DC
54Ø8:CØ E2 F3 4Ø D1 D1 A2 4Ø:22
541Ø:81 56 28 C1 C3 Ø3 BC C2:22
5418:84 ØC 5C 84 4C CC 98 4C:F8
542Ø:CC 3Ø FØ C4 3Ø 7Ø B4 C3:14
5428:Ø3 FC 3C 41 C3 Ø3 A8 ØØ:A8
543Ø:84 98 4Ø 4C 7Ø 4Ø 4C 2Ø:EC
5438:84 98 AØ 84 98 AØ 4Ø 4C:Ø8
544Ø:2Ø 4Ø 4C 7Ø ØØ 84 98 ØØ:18
5448:CØ ØØ 4Ø D1 8Ø 4Ø E2 A2:9E
545Ø:D1 D1 51 E2 F3 A2 51 A2:64
5458:ØØ 51 51 ØØ ØØ A2 ØØ 4Ø:ØD
546Ø:Ø8 84 98 ØC 98 4C 7Ø CC:EC
5468:7Ø 1Ø AØ 4Ø Ø8 84 98 58:54
547Ø:BØ 58 FØ D8 7Ø 1Ø AØ 4Ø:44
5478:Ø8 84 98 8Ø 1Ø Ø8 5Ø ØØ:54
548Ø:5Ø ØØ AØ 4Ø Ø8 ØØ 1Ø ØØ:9C
5488:1Ø ØØ 1Ø ØØ 5Ø 1Ø AØ 44:8C
549Ø:ØØ 8C 88 44 ØØ ØØ 1Ø ØØ:14
5498:5Ø 1Ø AØ ØØ C2 C9 8B 45:B1
54AØ:6F 22 91 6C 3Ø 78 24 D2:56
54A8:87 55 B6 28 FE BD 2C ØA:D7
54BØ:Ø5 ØE 5D Ø4 43 9F F9 E2:34
54B8:BE 3C AC BA 4F 4F 4C 76:57
54CØ:CD 6D 9B 46 A6 7Ø 57 6Ø:E3
54C8:2A 11 B2 38 AC BC 21 D4:A9
54DØ:CA 72 24 E5 Ø2 7Ø CF A8:65
54D8:FC DØ AC FD 4Ø 77 CC 59:26
54EØ:FA 3Ø CD 81 ØC 66 DE 4B:Ø3
54E8:B9 65 DØ 1Ø 39 E6 55 43:5B
54FØ:1A 92 37 1D 9D 6C 2C Ø8:A8
54F8:B7 EB 4Ø 86 AA 2E DE 41:31
55ØØ:A3 2B 19 9A BC FØ D2 E3:23
55Ø8:46 4Ø 62 ØC 77 2C 8F 7F:B5
551Ø:31 F3 F7 D8 F4 AE B1 57:28
5518:D4 ØE 2D 73 E4 ØB B9 17:2D
552Ø:FC 46 DE E5 Ø5 1E 7E Ø9:B2
5528:58 Ø9 BE 5A 3F C5 65 64:45
553Ø:CB C4 25 DB D7 E3 AD 77:AB
5538:FD FB ØE D1 17 E5 13 6A:94
554Ø:C5 13 7A A8 CB EE 1A FD:B7
5548:B3 11 EØ 57 7E Ø4 96 B9:DE
555Ø:DC 74 37 Ø9 4E 46 6B 4C:A9
```

```
5558:F0 20 8E EE 88 33 56 98:33
5560:F6 33 16 1E 88 18 18 21:B3
5568:6A B1 05 4E 5C 95 9C 0F:D6
5570:CF 05 D2 8F 71 AB BF 98:80
5578:CD 99 B8 5A CC C7 3B 2C:FF
5580:8D 73 97 BD 6D 95 EC 83:2C
5588:27 78 DE 86 25 E1 DF 1E:AE
5590:6C F3 B2 C3 F3 2A DC D3:B0
5598:CC F4 FE D9 23 07 BA D9:B6
55A0:42 99 15 9A 75 85 3C 3E:0B
55A8:DA 2F CA 0D 8F E5 38 01:80
55B0:36 AE 7D 05 C7 E7 09 DF:A6
55B8:65 FE A3 0B 62 0A 41 E3:88
55C0:25 89 B0 05 A0 CF 00 7F:62
55C8:3C 56 3C 71 0D BD C4 1D:70
55D0:E4 EB A2 71 7A 42 C4 E0:D3
55D8:6F 76 DE 16 95 10 CC AD:BF
55E0:61 F5 97 C5 16 8A BC D1:AF
55E8:DD 93 33 15 6C E1 B4 C0:7B
55F0:89 70 B3 EB 83 7A EB 49:93
55F8:69 E6 22 11 20 53 20 2A:EE
5600:01 4D CB 84 90 EA 6B 11:23
5608:6C 9A 49 D8 81 E4 89 23:ED
5610:0F 43 93 5B 90 44 55 CF:53
5618:B2 F8 66 29 D1 57 A9 AB:FC
5620:CE BD 61 14 4A BE 50 6C:A7
5628:0F DF D9 48 F2 57 BD FE:D5
5630:E3 A4 AE E2 40 EF 02 6E:9B
5638:9D 22 6D 6F 39 61 09 44:34
5640:F6 EE B9 AF 95 52 B8 53:C4
5648:B7 1F 47 CC 30 C4 3A 03:CE
5650:64 33 BC 9E A3 D3 EC 5C:91
5658:83 FA 73 E9 11 43 1C 4A:1D
5660:BF CD F0 B6 07 87 98 6C:8C
5668:88 D8 C9 BB DA A1 BB F8:22
5670:85 55 C6 4B 5C 08 19 40:1E
5678:B2 5E 76 20 92 AB 7C A9:A6
5680:39 48 A5 73 F5 7F 10 69:D5
5688:B0 3B 12 63 83 54 D2 90:8B
5690:F9 96 FB 09 88 B6 5E 4F:16
5698:38 90 7A 3B 0E 59 6D E2:07
56A0:53 DD 28 95 F0 FA C6 18:85
56A8:A8 A9 3D D1 C2 75 BC BE:8F
56B0:7B 45 29 CF DF 34 5D D6:90
56B8:06 B8 BD A0 E7 ED 82 CD:42
56C0:16 65 EF 0C D6 78 A5 72:04
56C8:6B D2 8B 8A BB 84 06 43:F7
56D0:CD C5 E6 1A 7C B9 47 00:4A
56D8:E3 24 16 71 53 1D 76 93:7E
56E0:74 33 0F E3 9B 97 D4 3D:0E
56E8:31 6D 47 F9 FC 9E 7A B0:78
56F0:1E 7A 1F FC C8 C2 8F 00:02
56F8:09 0C 81 AD AE 14 88 E6:6C
5700:CF 7B FD 31 96 C5 F7 16:D4
5708:BD 48 D8 F5 3A 50 66 CD:3C
5710:92 74 D3 61 87 0C BE CB:53
5718:1D CA F5 ED 5D 15 6F A5:33
5720:98 C5 19 AF 09 D2 6D F3:4C
5728:33 0E 3C 4C 25 52 88 C2:1F
5730:5A 8B 9B BA 66 31 D5 0F:1F
5738:E3 BB BD DF 16 D5 CC CC:FB
5740:7B 79 BC 1C 9C 2D B1 33:31
5748:5E 1E C9 69 8E 38 2D 75:31
5750:85 FB 44 E1 8D F5 8D 53:64
5758:51 39 5A F6 B1 7F 4C DE:CB
5760:92 D8 2E DD 43 B7 F5 4A:93
5768:06 FE EE 60 6B 5D ED C2:DB
5770:DA F5 2C 78 DC A0 E8 3C:33
5778:59 21 01 BE 58 C2 A9 7E:38
5780:39 D8 AF 7D 94 7A D2 49:87
5788:59 0D 72 D1 97 7E C2 E8:61
5790:D3 7C BF 09 E4 26 37 C8:2C
5798:04 CE 5D 3C 0B 66 5D 76:6C
57A0:25 E8 65 BC 98 70 2A 92:59
57A8:50 D0 04 12 1F 53 25 19:9B
57B0:27 DE 40 60 EF F5 C6 70:7D
57B8:D4 D4 F8 94 06 C5 44 AA:AB
57C0:AE B8 CA BC 80 F3 75 17:A0
57C8:50 2E 72 5B D0 75 E9 8B:12
57D0:55 85 6B AF 66 00 C1 DD:B0
57D8:25 52 3D 7D 88 13 9E 8A:5F
57E0:B0 F6 35 8C 6D C0 4E 57:1D
57E8:FD E7 B9 BD 30 99 D8 2F:0F
57F0:8D C4 E1 67 99 1B F3 B6:8F
```

```
57F8:A8 C5 9B D6 ED F0 2C CD:87
5800:D3 20 D3 20 D3 20 D3 20:08
5808:D3 20 D3 20 D3 20 D3 20:10
5810:D3 20 D3 20 D3 20 D3 20:18
5818:D3 20 D3 20 D3 60 D3 20:A0
5820:D3 20 D3 20 D3 20 D3 20:28
5828:D3 20 D3 20 D3 20 D3 20:30
5830:D3 20 D3 20 D3 20 D3 20:38
5838:D3 20 D3 20 D3 60 D3 20:C0
5840:D3 20 D3 20 D3 20 D3 20:48
5848:D3 20 D3 20 D3 20 D3 20:50
5850:D3 20 D3 20 D3 20 D3 20:58
5858:D3 20 D3 20 D3 60 D3 20:E0
5860:D3 20 D3 20 D3 20 D3 20:68
5868:D3 20 D3 20 D3 20 D3 20:70
5870:D3 20 D3 20 D3 20 D3 20:78
5878:D3 20 D3 20 D3 60 D3 20:00
5880:D3 20 D3 20 D3 20 D3 20:88
5888:D3 20 D3 20 D3 20 D3 20:90
5890:D3 20 D3 20 D3 20 D3 20:98
5898:D3 20 D3 20 D3 60 D3 20:20
58A0:D3 20 D3 20 D3 20 D3 20:A8
58A8:D3 20 D3 20 D3 20 D3 20:B0
58B0:D3 20 D3 20 D3 20 D3 20:B8
58B8:D3 20 D3 20 D3 60 D3 20:40
58C0:D3 20 D3 20 D3 20 D3 20:C8
58C8:D3 20 D3 20 D3 20 D3 20:D0
58D0:D3 20 D3 20 D3 20 D3 20:D8
58D8:D3 20 D3 20 D3 60 D3 20:60
58E0:D3 20 D3 20 D3 20 D3 20:E8
58E8:D3 20 D3 20 D3 20 D3 20:F0
58F0:D3 20 D3 20 D3 20 D3 20:F8
58F8:D3 20 D3 20 D3 60 D3 20:80
5900:D3 20 D3 20 D3 20 D3 20:09
5908:D3 20 D3 20 D3 20 D3 20:11
5910:D3 20 D3 20 D3 20 D3 20:19
5918:D3 20 D3 20 D3 60 D3 20:A1
5920:D3 20 D3 20 D3 20 D3 20:29
5928:D3 20 D3 20 D3 20 D3 20:31
5930:D3 20 D3 20 D3 20 D3 20:39
5938:D3 20 D3 20 D3 60 D3 20:C1
5940:D3 20 D3 20 D3 20 D3 20:49
5948:D3 20 D3 20 D3 20 D3 20:51
5950:D3 20 D3 20 D3 20 D3 20:59
5958:D3 20 D3 20 D3 60 D3 20:E1
5960:D3 20 D3 20 D3 20 D3 20:69
5968:D3 20 D3 20 D3 20 D3 20:71
5970:D3 20 D3 20 D3 20 D3 20:79
5978:D3 20 D3 20 D3 60 D3 20:01
5980:D3 20 D3 20 D3 20 D3 20:89
5988:D3 20 D3 20 D3 20 D3 20:91
5990:D3 20 D3 20 D3 20 D3 20:99
5998:D3 20 D3 20 D3 60 D3 20:21
59A0:D3 20 D3 20 D3 20 D3 20:A9
59A8:D3 20 D3 20 D3 20 D3 20:B1
59B0:D3 20 D3 20 D3 20 D3 20:B9
59B8:D3 20 D3 20 D3 60 D3 20:41
59C0:D3 20 D3 20 D3 20 D3 20:C9
59C8:D3 20 D3 20 D3 20 D3 20:D1
59D0:D3 20 D3 20 D3 20 D3 20:D9
59D8:D3 20 D3 20 D3 60 D3 20:61
59E0:D3 20 D3 20 D3 20 D3 20:E9
59E8:D3 20 D3 20 D3 20 D3 20:F1
59F0:D3 20 D3 20 D3 20 D3 20:F9
59F8:D3 20 D3 20 D3 60 D3 20:81
5A00:D3 20 13 21 D3 20 13 21:96
5A08:D3 20 13 21 D3 20 13 21:9E
5A10:D3 20 13 21 D3 20 13 21:A6
5A18:D3 20 13 21 D3 60 13 21:2E
5A20:D3 20 13 21 D3 20 13 21:B6
5A28:D3 20 13 21 D3 20 13 21:BE
5A30:D3 20 13 21 D3 20 13 21:C6
5A38:D3 20 13 21 D3 60 13 21:4E
5A40:D3 20 13 21 D3 20 13 21:D6
5A48:D3 20 13 21 D3 20 13 21:DE
5A50:D3 20 13 21 D3 20 13 21:E6
5A58:D3 20 13 21 D3 60 13 21:6E
5A60:D3 20 13 21 D3 20 13 21:F6
5A68:D3 20 13 21 D3 20 13 21:FE
5A70:D3 20 13 21 D3 20 13 21:06
5A78:D3 20 13 21 D3 60 13 21:8E
5A80:D3 20 13 21 D3 20 13 21:16
5A88:D3 20 13 21 D3 20 13 21:1E
5A90:D3 20 13 21 D3 20 13 21:26
```

```
5A98:D3 20 13 21 D3 60 13 21:AE
5AA0:D3 20 13 21 D3 20 13 21:36
5AA8:D3 20 13 21 D3 20 13 21:3E
5AB0:D3 20 13 21 D3 20 13 21:46
5AB8:D3 20 13 21 D3 60 13 21:CE
5AC0:D3 20 13 21 D3 20 13 21:56
5AC8:D3 20 13 21 D3 20 13 21:5E
5AD0:D3 20 13 21 D3 20 13 21:66
5AD8:D3 20 13 21 D3 60 13 21:EE
5AE0:D3 20 13 21 D3 20 13 21:76
5AE8:D3 20 13 21 D3 20 13 21:7E
5AF0:D3 20 13 21 D3 20 13 21:86
5AF8:D3 20 13 21 D3 60 13 21:0E
5B00:13 21 13 21 13 21 13 21:1F
5B08:13 21 13 21 13 21 13 21:27
5B10:13 21 13 21 13 21 13 21:2F
5B18:13 21 13 21 13 61 13 21:B7
5B20:13 21 13 21 13 21 13 21:3F
5B28:13 21 13 21 13 21 13 21:47
5B30:13 21 13 21 13 21 13 21:4F
5B38:13 21 13 21 13 61 13 21:D7
5B40:13 21 13 21 13 21 13 21:5F
5B48:13 21 13 21 13 21 13 21:67
5B50:13 21 13 21 13 21 13 21:6F
5B58:13 21 13 21 13 61 13 21:F7
5B60:13 21 13 21 13 21 13 21:7F
5B68:13 21 13 21 13 21 13 21:87
5B70:13 21 13 21 13 21 13 21:8F
5B78:13 21 13 21 13 61 13 21:17
5B80:13 21 13 21 13 21 13 21:9F
5B88:13 21 13 21 13 21 13 21:A7
5B90:13 21 13 21 13 21 13 21:AF
5B98:13 21 13 21 13 61 13 21:37
5BA0:13 21 13 21 13 21 13 21:BF
5BA8:13 21 13 21 13 21 13 21:C7
5BB0:13 21 13 21 13 21 13 21:CF
5BB8:13 21 13 21 13 61 13 21:57
5BC0:13 21 13 21 13 21 13 21:DF
5BC8:13 21 13 21 13 21 13 21:E7
5BD0:13 21 13 21 13 21 13 21:EF
5BD8:13 21 13 21 13 61 13 21:77
5BE0:13 21 13 21 13 21 13 21:FF
5BE8:13 21 13 21 13 21 13 21:07
5BF0:13 21 13 21 13 21 13 21:0F
5BF8:13 21 13 21 13 61 13 21:97
5C00:13 21 13 21 13 21 13 21:20
5C08:13 21 13 21 13 21 13 21:28
5C10:13 21 13 21 13 21 13 21:30
5C18:13 21 13 21 13 61 13 21:B8
5C20:13 21 13 21 13 21 13 21:40
5C28:13 21 13 21 13 21 13 21:48
5C30:13 21 13 21 13 21 13 21:50
5C38:13 21 13 21 13 61 13 21:D8
5C40:13 21 13 21 13 21 13 21:60
5C48:13 21 13 21 13 21 13 21:68
5C50:13 21 13 21 13 21 13 21:70
5C58:13 21 13 21 13 61 13 21:F8
5C60:13 21 13 21 13 21 13 21:80
5C68:13 21 13 21 13 21 13 21:88
5C70:13 21 13 21 13 21 13 21:90
5C78:13 21 13 21 13 61 13 21:18
5C80:13 21 13 21 13 21 13 21:A0
5C88:13 21 13 21 13 21 13 21:A8
5C90:13 21 13 21 13 21 13 21:B0
5C98:13 21 13 21 13 61 13 21:38
5CA0:13 21 13 21 13 21 13 21:C0
5CA8:13 21 13 21 13 21 13 21:C8
5CB0:13 21 13 21 13 21 13 21:D0
5CB8:13 21 13 21 13 61 13 21:58
5CC0:13 21 13 21 13 21 13 21:E0
5CC8:13 21 13 21 13 21 13 21:E8
5CD0:13 21 13 21 13 21 13 21:F0
5CD8:13 21 13 21 13 61 13 21:78
5CE0:13 21 13 21 13 21 13 21:00
5CE8:13 21 13 21 13 21 13 21:08
5CF0:13 21 13 21 13 21 13 21:10
5CF8:13 21 13 21 13 61 13 A1:98
5D00:53 21 53 21 53 21 53 21:21
5D08:53 21 53 21 53 21 53 21:29
5D10:53 21 53 21 53 21 53 21:31
5D18:53 61 53 21 53 21 53 21:B9
5D20:53 21 53 21 53 21 53 21:41
5D28:53 21 53 21 53 21 53 21:49
5D30:53 21 53 21 53 61 53 21:D1
```

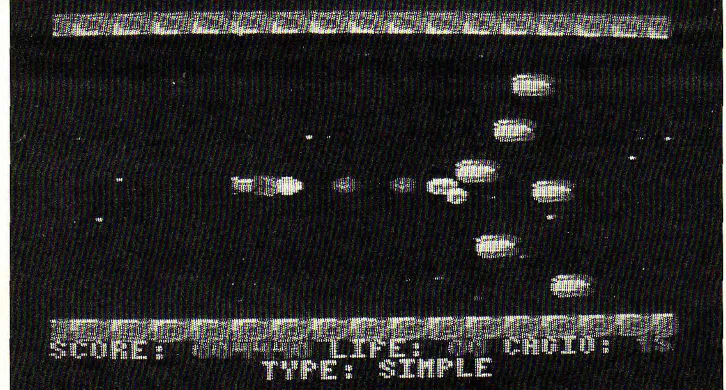
```
5D38:53 21 53 21 53 21 53 21:59
5D40:53 21 53 21 53 21 53 21:61
5D48:53 21 53 21 53 21 53 21:69
5D50:53 61 53 21 53 21 53 21:F1
5D58:53 21 53 21 53 21 53 21:79
5D60:53 21 53 21 53 21 53 21:81
5D68:53 21 53 21 53 61 53 21:09
5D70:53 21 53 21 53 21 53 21:91
5D78:53 21 53 21 53 21 53 21:99
5D80:53 21 53 21 53 21 53 21:A1
5D88:53 61 53 21 53 21 53 21:29
5D90:53 21 53 21 53 21 53 21:B1
5D98:53 21 53 21 53 21 53 21:B9
5DA0:53 21 53 21 53 61 53 21:41
5DA8:53 21 53 21 53 21 53 21:C9
5DB0:53 21 53 21 53 21 53 21:D1
5DB8:53 21 53 21 53 21 53 21:D9
5DC0:53 61 53 21 53 21 53 21:61
5DC8:53 21 53 21 53 21 53 21:E9
5DD0:53 21 53 21 53 21 53 21:F1
5DD8:53 21 53 21 53 61 53 21:79
5DE0:53 21 53 21 53 21 53 21:01
5DE8:53 21 53 21 53 21 53 21:09
5DF0:53 21 53 21 53 21 53 21:11
5DF8:53 61 53 21 53 21 53 21:99
5E00:53 21 53 21 53 21 53 21:22
5E08:53 21 53 21 53 21 53 21:2A
5E10:53 21 53 21 53 61 53 21:B2
5E18:53 21 53 21 53 21 53 21:3A
5E20:53 21 53 21 53 21 53 21:42
5E28:53 21 53 21 53 21 53 21:4A
5E30:53 61 53 21 53 21 53 21:D2
5E38:53 21 53 21 53 21 53 21:5A
5E40:53 21 53 21 53 21 53 21:62
5E48:53 21 53 21 53 61 53 21:EA
5E50:53 21 53 21 53 21 53 21:72
5E58:53 21 53 21 53 21 53 21:7A
5E60:53 21 53 21 53 21 53 21:82
5E68:53 61 53 21 53 21 53 21:0A
5E70:53 21 53 21 53 21 53 21:92
5E78:53 21 53 21 53 21 53 21:9A
5E80:53 21 53 21 53 61 53 21:22
5E88:53 21 53 21 53 21 53 21:AA
5E90:53 21 53 21 53 21 53 21:B2
5E98:53 21 53 21 53 21 53 21:BA
5EA0:53 61 53 21 53 21 53 21:42
5EA8:53 21 53 21 53 21 53 21:CA
5EB0:53 21 53 21 53 21 53 21:D2
5EB8:53 21 53 21 53 E1 53 21:5A
5EC0:93 21 93 21 93 21 93 21:E2
5EC8:93 21 93 21 93 21 93 21:EA
5ED0:93 21 93 21 93 21 93 21:F2
5ED8:93 61 93 21 93 21 93 21:7A
5EE0:93 21 93 21 93 21 93 21:02
5EE8:93 21 93 21 93 21 93 21:0A
5EF0:93 21 93 21 93 61 93 21:92
5EF8:93 21 93 21 93 21 93 21:1A
5F00:93 21 93 21 93 21 93 21:23
5F08:93 21 93 21 93 21 93 21:2B
5F10:93 61 93 21 93 21 93 21:B3
5F18:93 21 93 21 93 21 93 21:3B
5F20:93 21 93 21 93 21 93 21:43
5F28:93 21 93 21 93 61 93 21:CB
5F30:93 21 93 21 93 21 93 21:53
5F38:93 21 93 21 93 21 93 21:5B
5F40:93 21 93 21 93 21 93 21:63
5F48:93 61 93 21 93 21 93 21:EB
5F50:93 21 93 21 93 21 93 21:73
5F58:93 21 93 21 93 21 93 21:7B
5F60:93 21 93 21 93 61 93 21:03
5F68:93 21 93 21 93 21 93 21:8B
5F70:93 21 93 21 93 21 93 21:93
5F78:93 21 93 21 93 21 93 21:9B
5F80:93 61 93 21 93 21 93 21:23
5F88:93 21 93 21 93 21 93 21:AB
5F90:93 21 93 21 93 21 93 21:B3
5F98:93 21 93 21 93 61 93 21:3B
5FA0:93 21 93 21 93 21 93 21:C3
5FA8:93 21 93 21 93 21 93 21:CB
5FB0:93 21 93 21 93 21 93 21:D3
5FB8:93 61 93 21 93 21 93 21:5B
5FC0:93 21 93 21 93 21 93 21:E3
5FC8:93 21 93 21 93 21 93 21:EB
5FD0:93 21 93 21 93 61 93 21:73
5FD8:93 21 93 21 93 21 93 21:FB
5FE0:93 21 93 21 93 21 93 21:03
5FE8:93 21 93 21 93 21 93 21:0B
5FF0:93 61 93 21 93 21 93 21:93
5FF8:93 21 93 21 93 21 93 21:1B
6000:93 21 93 21 93 21 93 21:24
6008:93 21 93 21 93 61 93 21:AC
6010:93 21 93 21 93 21 93 21:34
6018:93 21 93 21 93 21 93 21:3C
6020:93 21 93 21 93 21 93 21:44
6028:93 61 93 21 93 21 93 21:CC
6030:93 21 93 21 93 21 93 21:54
6038:93 21 93 21 93 21 93 21:5C
6040:93 21 93 21 93 61 93 21:E4
6048:93 21 93 21 93 21 93 21:6C
6050:93 21 93 21 93 21 93 21:74
6058:93 21 93 21 93 21 93 21:7C
6060:93 61 93 21 93 21 93 21:04
6068:93 21 93 21 93 21 93 21:8C
6070:93 21 93 21 93 21 93 21:94
6078:93 21 93 21 93 61 93 21:1C
6080:93 21 93 21 93 21 93 21:A4
6088:93 21 93 21 93 21 93 21:AC
6090:93 21 93 21 93 21 93 21:B4
6098:93 61 93 21 93 21 93 21:3C
60A0:93 21 93 21 93 21 93 21:C4
60A8:93 21 93 21 93 21 93 21:CC
60B0:93 21 93 21 93 61 93 21:54
60B8:93 21 93 21 93 21 93 21:DC
60C0:93 21 93 21 93 21 93 21:E4
60C8:93 21 93 21 93 21 93 21:EC
60D0:93 61 93 21 93 21 93 21:74
60D8:93 21 93 21 93 21 93 21:FC
60E0:93 21 93 21 93 21 93 21:04
60E8:93 21 93 21 93 61 93 21:8C
60F0:93 21 93 21 93 21 93 21:14
60F8:93 21 93 21 93 21 93 21:1C
6100:93 21 93 21 93 21 93 21:25
6108:93 61 93 21 93 21 93 21:AD
6110:93 21 93 21 93 21 93 21:35
6118:93 21 93 21 93 21 93 21:3D
6120:93 21 93 21 93 61 93 21:C5
6128:93 21 93 21 93 21 93 21:4D
6130:93 21 93 21 93 21 93 21:55
6138:93 21 93 21 93 21 93 21:5D
6140:93 61 93 21 93 21 93 21:E5
6148:93 21 93 21 93 21 93 21:6D
6150:93 21 93 21 93 21 93 21:75
6158:93 21 93 21 93 E1 93 21:FD
6160:D3 21 D3 21 D3 21 D3 21:85
6168:D3 21 D3 21 D3 21 D3 21:8D
6170:D3 21 D3 21 D3 61 D3 21:15
6178:D3 21 D3 21 D3 21 D3 21:9D
6180:D3 21 D3 21 D3 21 D3 21:A5
6188:D3 21 D3 21 D3 61 D3 21:2D
6190:D3 21 D3 21 D3 21 D3 21:B5
6198:D3 21 D3 21 D3 21 D3 21:BD
61A0:D3 21 D3 21 D3 61 D3 21:45
61A8:D3 21 D3 21 D3 21 D3 21:CD
61B0:D3 21 D3 21 D3 21 D3 21:D5
61B8:D3 21 D3 21 D3 61 D3 21:5D
61C0:D3 21 D3 21 D3 21 D3 21:E5
61C8:D3 21 D3 21 D3 21 D3 21:ED
61D0:D3 21 D3 21 D3 61 D3 21:75
61D8:D3 21 D3 21 D3 21 D3 21:FD
61E0:D3 21 D3 21 D3 21 D3 21:05
61E8:D3 21 D3 21 D3 61 D3 21:8D
61F0:D3 21 D3 21 D3 21 D3 21:15
61F8:D3 21 D3 21 D3 21 D3 21:1D
6200:D3 21 D3 21 D3 61 D3 21:A6
6208:D3 21 D3 21 D3 21 D3 21:2E
6210:D3 21 D3 21 D3 21 D3 21:36
6218:D3 21 D3 21 D3 61 D3 21:BE
6220:D3 21 D3 21 D3 21 D3 21:46
6228:D3 21 D3 21 D3 21 D3 21:4E
6230:D3 21 D3 21 D3 61 D3 21:D6
6238:D3 21 D3 21 D3 21 D3 21:5E
6240:D3 21 D3 21 D3 21 D3 21:66
6248:D3 21 D3 21 D3 61 D3 21:EE
6250:D3 21 D3 21 D3 21 D3 21:76
6258:D3 21 D3 21 D3 21 D3 21:7E
6260:D3 21 D3 21 D3 61 D3 21:06
6268:D3 21 D3 21 D3 21 D3 21:8E
6270:D3 21 D3 21 D3 21 D3 21:96
6278:D3 21 D3 21 D3 61 D3 21:1E
6280:D3 21 D3 21 D3 21 D3 21:A6
6288:D3 21 D3 21 D3 21 D3 21:AE
6290:D3 21 D3 21 D3 61 D3 21:36
6298:D3 21 D3 21 D3 21 D3 21:BE
62A0:D3 21 D3 21 D3 21 D3 21:C6
62A8:D3 21 D3 21 D3 61 D3 21:4E
62B0:D3 21 D3 21 D3 21 D3 21:D6
62B8:D3 21 D3 21 D3 21 D3 21:DE
62C0:D3 21 D3 21 D3 61 D3 21:66
62C8:D3 21 D3 21 D3 21 D3 21:EE
62D0:D3 21 D3 21 D3 21 D3 21:F6
62D8:D3 21 D3 21 D3 61 D3 21:7E
62E0:D3 21 D3 21 D3 21 D3 21:06
62E8:D3 21 D3 21 D3 21 D3 21:0E
62F0:D3 21 D3 21 D3 61 D3 21:96
62F8:D3 21 D3 21 D3 21 D3 21:1E
6300:D3 21 D3 21 D3 21 D3 21:27
6308:D3 21 D3 21 D3 61 D3 21:AF
6310:D3 21 D3 21 D3 21 D3 21:37
6318:D3 21 D3 21 D3 21 D3 21:3F
6320:D3 21 D3 21 D3 61 D3 21:C7
6328:D3 21 D3 21 D3 21 D3 21:4F
6330:D3 21 D3 21 D3 21 D3 21:57
6338:D3 21 D3 21 D3 61 D3 21:DF
6340:D3 21 D3 21 D3 21 D3 21:67
6348:D3 21 D3 21 D3 21 D3 21:6F
6350:D3 21 D3 21 D3 61 D3 21:F7
6358:D3 21 D3 21 D3 21 D3 21:7F
6360:D3 21 D3 21 D3 21 D3 21:87
6368:D3 21 D3 21 D3 61 D3 21:0F
6370:D3 21 D3 21 D3 21 D3 21:97
6378:D3 21 D3 21 D3 21 D3 21:9F
6380:D3 21 D3 21 D3 61 D3 21:27
6388:D3 21 D3 21 D3 21 D3 21:AF
6390:D3 21 D3 21 D3 21 D3 21:B7
6398:D3 21 D3 21 D3 61 D3 A1:3F
63A0:13 22 13 22 13 22 13 22:DB
63A8:13 22 13 22 13 62 13 22:63
63B0:13 22 13 22 13 22 13 22:EB
63B8:13 22 13 22 13 62 13 22:73
63C0:13 22 13 22 13 22 13 22:FB
63C8:13 22 13 22 13 62 13 22:83
63D0:13 22 13 22 13 22 13 22:0B
63D8:13 22 13 22 13 62 13 22:93
63E0:13 22 13 22 13 22 13 22:1B
63E8:13 22 13 22 13 62 13 22:A3
63F0:13 22 13 22 13 22 13 22:2B
63F8:13 22 13 22 13 62 13 22:B3
6400:13 22 13 22 13 22 13 22:3C
6408:13 22 13 22 13 62 13 22:C4
6410:13 22 13 22 13 22 13 22:4C
6418:13 22 13 22 13 62 13 22:D4
6420:13 22 13 22 13 22 13 22:5C
6428:13 22 13 22 13 62 13 22:E4
6430:13 22 13 22 13 22 13 22:6C
6438:13 22 13 22 13 62 13 22:F4
6440:13 22 13 22 13 22 13 22:7C
6448:13 22 13 22 13 62 13 22:04
6450:13 22 13 22 13 22 13 22:8C
6458:13 22 13 22 13 62 13 22:14
6460:13 22 13 22 13 22 13 22:9C
6468:13 22 13 22 13 62 13 22:24
6470:13 22 13 22 13 22 13 22:AC
6478:13 22 13 22 13 62 13 22:34
6480:13 22 13 22 13 22 13 22:BC
6488:13 22 13 22 13 62 13 22:44
6490:13 22 13 22 13 22 13 22:CC
6498:13 22 13 22 13 62 13 22:54
64A0:13 22 13 22 13 22 13 22:DC
64A8:13 22 13 22 13 62 13 22:64
64B0:13 22 13 22 13 22 13 22:EC
64B8:13 22 13 22 13 62 13 22:74
64C0:13 22 13 22 13 22 13 22:FC
64C8:13 22 13 22 13 62 13 22:84
64D0:13 22 13 22 13 22 13 22:0C
64D8:13 22 13 22 13 62 13 22:94
64E0:13 22 13 22 13 22 13 22:1C
64E8:13 22 13 22 13 62 13 22:A4
64F0:13 22 13 22 13 22 13 22:2C
64F8:13 22 13 22 13 62 13 22:B4
6500:13 22 13 22 13 22 13 22:3D
6508:13 22 13 22 13 62 13 22:C5
6510:13 22 13 22 13 22 13 22:4D
```

```
6518:13 22 13 22 13 62 13 22:D5
6520:13 22 13 22 13 22 13 22:5D
6528:13 22 13 22 13 62 13 22:E5
6530:13 22 13 22 13 22 13 22:6D
6538:13 22 13 22 13 62 13 22:F5
6540:13 22 13 22 13 22 13 22:7D
6548:13 22 13 22 13 62 13 22:05
6550:13 22 13 22 13 22 13 22:8D
6558:13 22 13 22 13 62 13 22:15
6560:13 22 13 22 13 22 13 22:9D
6568:13 22 13 22 13 62 13 22:25
6570:13 22 13 22 13 22 13 22:AD
6578:13 22 13 22 13 62 13 22:35
6580:13 22 13 22 13 22 13 22:BD
6588:13 22 13 22 13 62 13 22:45
6590:13 22 13 22 13 22 13 22:CD
6598:13 22 13 22 13 62 13 22:55
65A0:13 22 13 22 13 22 13 22:DD
65A8:13 22 13 22 13 62 13 22:65
65B0:13 22 13 22 13 22 13 22:ED
65B8:13 22 13 22 13 62 13 22:75
65C0:13 22 13 22 13 22 13 22:FD
65C8:13 22 13 22 13 62 13 22:85
65D0:13 22 13 22 13 22 13 22:0D
65D8:13 22 13 22 13 62 13 22:95
65E0:13 22 13 22 13 22 13 22:1D
65E8:13 22 13 22 13 62 13 22:A5
65F0:13 22 13 22 13 22 13 22:2D
65F8:13 22 13 22 13 62 13 22:B5
6600:13 22 13 22 13 22 13 22:3E
6608:13 22 13 22 13 62 13 22:C6
6610:13 22 13 22 13 22 13 22:4E
6618:13 22 13 22 13 E2 13 22:D6
6620:53 22 53 22 53 22 53 22:5E
6628:53 22 53 22 53 62 53 22:E6
6630:53 22 53 22 53 22 53 22:6E
6638:53 22 53 22 53 62 53 22:F6
6640:53 22 53 22 53 22 53 22:7E
6648:53 22 53 22 53 62 53 22:06
6650:53 22 53 22 53 22 53 22:8E
6658:53 22 53 22 53 62 53 22:16
6660:53 22 53 22 53 22 53 22:9E
6668:53 22 53 22 53 62 53 22:26
6670:53 22 53 22 53 22 53 22:AE
6678:53 22 53 22 53 62 53 22:36
6680:53 22 53 22 53 22 53 22:BE
6688:53 22 53 22 53 62 53 22:46
6690:53 22 53 22 53 22 53 22:CE
6698:53 22 53 22 53 62 53 22:56
66A0:53 22 53 22 53 22 53 22:DE
66A8:53 22 53 22 53 62 53 22:66
66B0:53 22 53 22 53 22 53 22:EE
66B8:53 22 53 22 53 62 53 22:76
66C0:53 22 53 22 53 22 53 22:FE
66C8:53 22 53 22 53 62 53 22:86
66D0:53 22 53 22 53 22 53 22:0E
66D8:53 22 53 22 53 62 53 22:96
66E0:53 22 53 22 53 22 53 22:1E
66E8:53 22 53 22 53 62 53 22:A6
66F0:53 22 53 22 53 22 53 22:2E
66F8:53 22 53 22 53 62 53 22:B6
6700:53 22 53 22 53 22 53 22:3F
6708:53 22 53 22 53 62 53 22:C7
6710:53 22 53 22 53 22 53 22:4F
6718:53 22 53 22 53 62 53 22:D7
6720:53 22 53 22 53 22 53 22:5F
6728:53 22 53 22 53 62 53 22:E7
6730:53 22 53 22 53 22 53 22:6F
6738:53 22 53 22 53 62 53 22:F7
6740:53 22 53 22 53 22 53 22:7F
6748:53 22 53 22 53 62 53 22:07
6750:53 22 53 22 53 22 53 22:8F
6758:53 22 53 22 53 62 53 22:17
6760:53 22 53 22 53 22 53 22:9F
6768:53 22 53 22 53 62 53 22:27
6770:53 22 53 22 53 22 53 22:AF
6778:53 22 53 22 53 62 53 22:37
6780:53 22 53 22 53 22 53 22:BF
6788:53 22 53 22 53 62 53 22:47
6790:53 22 53 22 53 22 53 22:CF
6798:53 22 53 22 53 62 53 22:57
67A0:53 22 53 22 53 22 53 22:DF
67A8:53 22 53 22 53 62 53 22:67
67B0:53 22 53 22 53 22 53 22:EF
67B8:53 22 53 22 53 62 53 22:77
67C0:53 22 53 22 53 22 53 22:FF
67C8:53 22 53 22 53 62 53 22:87
67D0:53 22 53 22 53 22 53 22:0F
67D8:53 22 53 22 53 62 53 22:97
67E0:53 22 53 22 53 22 53 22:1F
67E8:53 22 53 22 53 62 53 22:A7
67F0:53 22 53 22 53 22 53 22:2F
67F8:53 22 53 22 53 62 53 22:B7
6800:53 22 53 22 53 22 53 22:40
6808:53 22 53 22 53 62 53 22:C8
6810:53 22 53 22 53 22 53 22:50
6818:53 22 53 22 53 62 53 22:D8
6820:53 22 53 22 53 22 53 22:60
6828:53 22 53 22 53 62 53 22:E8
6830:53 22 53 22 53 22 53 22:70
6838:53 22 53 22 53 62 53 22:F8
6840:53 22 53 22 53 22 53 22:80
6848:53 22 53 22 53 62 53 22:08
6850:53 22 53 22 53 22 53 22:90
6858:53 22 53 22 53 62 53 22:18
6860:53 22 53 22 53 22 53 22:A0
6868:53 22 53 22 53 62 53 22:28
6870:53 22 53 22 53 22 53 22:B0
6878:53 22 53 22 53 62 53 22:38
6880:53 22 53 22 53 22 53 22:C0
6888:53 22 53 22 53 62 53 22:48
6890:53 22 53 22 53 22 53 22:D0
6898:53 22 53 22 53 62 53 A2:58
68A0:93 22 93 22 93 22 93 22:E0
68A8:93 22 93 22 93 22 93 22:E8
68B0:93 62 93 22 93 22 93 22:70
68B8:93 22 93 22 93 22 93 22:F8
68C0:93 22 93 22 93 62 93 22:80
68C8:93 22 93 22 93 22 93 22:08
68D0:93 22 93 22 93 22 93 22:10
68D8:93 62 93 22 93 22 93 22:98
68E0:93 22 93 22 93 22 93 22:20
68E8:93 22 93 22 93 62 93 22:A8
68F0:93 22 93 22 93 22 93 22:30
68F8:93 22 93 22 93 22 93 22:38
6900:93 62 93 22 93 22 93 22:C1
6908:93 22 93 22 93 22 93 22:49
6910:93 22 93 22 93 62 93 22:D1
6918:93 22 93 22 93 22 93 22:59
6920:93 22 93 22 93 22 93 22:61
6928:93 62 93 22 93 22 93 22:E9
6930:93 22 93 22 93 22 93 22:71
6938:93 22 93 22 93 62 93 22:F9
6940:93 22 93 22 93 22 93 22:81
6948:93 22 93 22 93 22 93 22:89
6950:93 62 93 22 93 22 93 22:11
6958:93 22 93 22 93 22 93 22:99
6960:93 22 93 22 93 62 93 22:21
6968:93 22 93 22 93 22 93 22:A9
6970:93 22 93 22 93 22 93 22:B1
6978:93 62 93 22 93 22 93 22:39
6980:93 22 93 22 93 22 93 22:C1
6988:93 22 93 22 93 62 93 22:49
6990:93 22 93 22 93 22 93 22:D1
6998:93 22 93 22 93 22 93 22:D9
69A0:93 62 93 22 93 22 93 22:61
69A8:93 22 93 22 93 22 93 22:E9
69B0:93 22 93 22 93 62 93 22:71
69B8:93 22 93 22 93 22 93 22:F9
69C0:93 22 93 22 93 22 93 22:01
69C8:93 62 93 22 93 22 93 22:89
69D0:93 22 93 22 93 22 93 22:11
69D8:93 22 93 22 93 62 93 22:99
69E0:93 22 93 22 93 22 93 22:21
69E8:93 22 93 22 93 22 93 22:29
69F0:93 62 93 22 93 22 93 22:B1
69F8:93 22 93 22 93 22 93 22:39
6A00:93 22 93 22 93 62 93 22:C2
6A08:93 22 93 22 93 22 93 22:4A
6A10:93 22 93 22 93 22 93 22:52
6A18:93 62 93 22 93 22 93 22:DA
6A20:93 22 93 22 93 22 93 22:62
6A28:93 22 93 22 93 62 93 22:EA
6A30:93 22 93 22 93 22 93 22:72
6A38:93 22 93 22 93 22 93 22:7A
6A40:93 62 93 22 93 22 93 22:02
6A48:93 22 93 22 93 22 93 22:8A
6A50:93 22 93 22 93 62 93 22:12
6A58:93 22 93 22 93 22 93 22:9A
6A60:93 22 93 22 93 22 93 22:A2
6A68:93 62 93 22 93 22 93 22:2A
6A70:93 22 93 22 93 22 93 22:B2
6A78:93 22 93 22 93 62 93 22:3A
6A80:93 22 93 22 93 22 93 22:C2
6A88:93 22 93 22 93 22 93 22:CA
6A90:93 62 93 22 93 22 93 22:52
6A98:93 22 93 22 93 22 93 22:DA
6AA0:93 22 93 22 93 62 93 22:62
6AA8:93 22 93 22 93 22 93 22:EA
6AB0:93 22 93 22 93 22 93 22:F2
6AB8:93 62 93 22 93 22 93 22:7A
6AC0:93 22 93 22 93 22 93 22:02
6AC8:93 22 93 22 93 62 93 22:8A
6AD0:93 22 93 22 93 22 93 22:12
6AD8:93 22 93 22 93 22 93 22:1A
6AE0:93 62 93 22 93 22 93 22:A2
6AE8:93 22 93 22 93 22 93 22:2A
6AF0:93 22 93 22 93 62 93 22:B2
6AF8:93 22 93 22 93 22 93 22:3A
6B00:93 22 93 22 93 22 93 22:43
6B08:93 62 93 22 93 22 93 22:CB
6B10:93 22 93 22 93 22 93 22:53
6B18:93 22 93 22 93 E2 93 22:DB
6B20:D3 22 D3 22 D3 22 D3 22:63
6B28:D3 22 D3 22 D3 22 D3 22:6B
6B30:D3 22 D3 22 D3 62 D3 22:F3
6B38:D3 22 D3 22 D3 22 D3 22:7B
6B40:D3 22 D3 22 D3 22 D3 22:83
6B48:D3 22 D3 22 D3 62 D3 22:0B
6B50:D3 22 D3 22 D3 22 D3 22:93
6B58:D3 22 D3 22 D3 22 D3 22:9B
6B60:D3 22 D3 22 D3 62 D3 22:23
6B68:D3 22 D3 22 D3 22 D3 22:AB
6B70:D3 22 D3 22 D3 22 D3 22:B3
6B78:D3 22 D3 22 D3 62 D3 22:3B
6B80:D3 22 D3 22 D3 22 D3 22:C3
6B88:D3 22 D3 22 D3 22 D3 22:CB
6B90:D3 22 D3 22 D3 62 D3 22:53
6B98:D3 22 D3 22 D3 22 D3 22:DB
6BA0:D3 22 D3 22 D3 22 D3 22:E3
6BA8:D3 22 D3 22 D3 62 D3 22:6B
6BB0:D3 22 D3 22 D3 22 D3 22:F3
6BB8:D3 22 D3 22 D3 22 D3 22:FB
6BC0:D3 22 D3 22 D3 62 D3 22:83
6BC8:D3 22 D3 22 D3 22 D3 22:0B
6BD0:D3 22 D3 22 D3 22 D3 22:13
6BD8:D3 22 D3 22 D3 62 D3 22:9B
6BE0:D3 22 D3 22 D3 22 D3 22:23
6BE8:D3 22 D3 22 D3 22 D3 22:2B
6BF0:D3 22 D3 22 D3 62 D3 22:B3
6BF8:D3 22 D3 22 D3 22 D3 22:3B
6C00:D3 22 D3 22 D3 22 D3 22:44
6C08:D3 22 D3 22 D3 62 D3 22:CC
6C10:D3 22 D3 22 D3 22 D3 22:54
6C18:D3 22 D3 22 D3 22 D3 22:5C
6C20:D3 22 D3 22 D3 62 D3 22:E4
6C28:D3 22 D3 22 D3 22 D3 22:6C
6C30:D3 22 D3 22 D3 22 D3 22:74
6C38:D3 22 D3 22 D3 62 D3 22:FC
6C40:D3 22 D3 22 D3 22 D3 22:84
6C48:D3 22 D3 22 D3 22 D3 22:8C
6C50:D3 22 D3 22 D3 62 D3 22:14
6C58:D3 22 D3 22 D3 22 D3 22:9C
6C60:D3 22 D3 22 D3 22 D3 22:A4
6C68:D3 22 D3 22 D3 62 D3 22:2C
6C70:D3 22 D3 22 D3 22 D3 22:B4
6C78:D3 22 D3 22 D3 22 D3 22:BC
6C80:D3 22 D3 22 D3 62 D3 22:44
6C88:D3 22 D3 22 D3 22 D3 22:CC
6C90:D3 22 D3 22 D3 22 D3 22:D4
6C98:D3 22 D3 22 D3 62 D3 22:5C
6CA0:D3 22 D3 22 D3 22 D3 22:E4
6CA8:D3 22 D3 22 D3 22 D3 22:EC
6CB0:D3 22 D3 22 D3 62 D3 22:74
6CB8:D3 22 D3 22 D3 22 D3 22:FC
6CC0:D3 22 D3 22 D3 22 D3 22:04
6CC8:D3 22 D3 22 D3 62 D3 22:8C
6CD0:D3 22 D3 22 D3 22 D3 22:14
6CD8:D3 22 D3 22 D3 22 D3 22:1C
6CE0:D3 22 D3 22 D3 62 D3 22:A4
6CE8:D3 22 D3 22 D3 22 D3 22:2C
6CF0:D3 22 D3 22 D3 22 D3 22:34
```

```
6CF8:D3 22 D3 22 D3 62 D3 22:BC
6D00:D3 22 D3 22 D3 22 D3 22:45
6D08:D3 22 D3 22 D3 22 D3 22:4D
6D10:D3 22 D3 22 D3 62 D3 22:D5
6D18:D3 22 D3 22 D3 22 D3 22:5D
6D20:D3 22 D3 22 D3 22 D3 22:65
6D28:D3 22 D3 22 D3 62 D3 22:ED
6D30:D3 22 D3 22 D3 22 D3 22:75
6D38:D3 22 D3 22 D3 22 D3 22:7D
6D40:D3 22 D3 22 D3 62 D3 22:05
6D48:D3 22 D3 22 D3 22 D3 22:8D
6D50:D3 22 D3 22 D3 22 D3 22:95
6D58:D3 22 D3 22 D3 62 D3 A2:1D
6D60:13 23 13 23 13 23 13 23:B9
6D68:13 23 13 23 13 23 13 23:C1
6D70:13 23 13 23 13 23 13 23:C9
6D78:13 63 13 23 13 23 13 23:51
6D80:13 23 13 23 13 23 13 23:D9
6D88:13 23 13 23 13 23 13 23:E1
6D90:13 23 13 23 13 63 13 23:69
6D98:13 23 13 23 13 23 13 23:F1
6DA0:13 23 13 23 13 23 13 23:F9
6DA8:13 23 13 23 13 23 13 23:01
6DB0:13 63 13 23 13 23 13 23:89
6DB8:13 23 13 23 13 23 13 23:11
6DC0:13 23 13 23 13 23 13 23:19
6DC8:13 23 13 23 13 63 13 23:A1
6DD0:13 23 13 23 13 23 13 23:29
6DD8:13 23 13 23 13 23 13 23:31
6DE0:13 23 13 23 13 23 13 23:39
6DE8:13 63 13 23 13 23 13 23:C1
6DF0:13 23 13 23 13 23 13 23:49
6DF8:13 23 13 23 13 23 13 23:51
6E00:13 23 13 23 13 63 13 23:DA
6E08:13 23 13 23 13 23 13 23:62
6E10:13 23 13 23 13 23 13 23:6A
6E18:13 23 13 23 13 23 13 23:72
6E20:13 63 13 23 13 23 13 23:FA
6E28:13 23 13 23 13 23 13 23:82
6E30:13 23 13 23 13 23 13 23:8A
6E38:13 23 13 23 13 63 13 23:12
6E40:13 23 13 23 13 23 13 23:9A
6E48:13 23 13 23 13 23 13 23:A2
6E50:13 23 13 23 13 23 13 23:AA
6E58:13 63 13 23 13 23 13 23:32
6E60:13 23 13 23 13 23 13 23:BA
6E68:13 23 13 23 13 23 13 23:C2
6E70:13 23 13 23 13 63 13 23:4A
6E78:13 23 13 23 13 23 13 23:D2
6E80:13 23 13 23 13 23 13 23:DA
6E88:13 23 13 23 13 23 13 23:E2
6E90:13 63 13 23 13 23 13 23:6A
6E98:13 23 13 23 13 23 13 23:F2
6EA0:13 23 13 23 13 23 13 23:FA
6EA8:13 23 13 23 13 63 13 23:82
6EB0:13 23 13 23 13 23 13 23:0A
6EB8:13 23 13 23 13 23 13 23:12
6EC0:13 23 13 23 13 23 13 23:1A
6EC8:13 63 13 23 13 23 13 23:A2
6ED0:13 23 13 23 13 23 13 23:2A
6ED8:13 23 13 23 13 23 13 23:32
6EE0:13 23 13 23 13 63 13 23:BA
6EE8:13 23 13 23 13 23 13 23:42
6EF0:13 23 13 23 13 23 13 23:4A
6EF8:13 23 13 23 13 23 13 23:52
6F00:13 63 13 23 13 23 13 23:DB
6F08:13 23 13 23 13 23 13 23:63
6F10:13 23 13 23 13 23 13 23:6B
6F18:13 23 13 23 13 63 13 23:F3
6F20:13 23 13 23 13 23 13 23:7B
6F28:13 23 13 23 13 23 13 23:83
6F30:13 23 13 23 13 23 13 23:8B
6F38:13 63 13 23 13 23 13 23:13
6F40:13 23 13 23 13 23 13 23:9B
6F48:13 23 13 23 13 23 13 23:A3
6F50:13 23 13 23 13 63 13 23:2B
6F58:13 23 13 23 13 23 13 23:B3
6F60:13 23 13 23 13 23 13 23:BB
6F68:13 23 13 23 13 23 13 23:C3
6F70:13 63 13 23 13 23 13 23:4B
6F78:13 23 13 23 13 23 13 23:D3
6F80:13 23 13 23 13 23 13 23:DB
6F88:13 23 13 23 13 63 13 23:63
6F90:13 23 13 23 13 23 13 23:EB
6F98:13 23 13 23 13 23 13 23:F3
6FA0:13 23 13 23 13 23 13 23:FB
6FA8:13 63 13 23 13 23 13 23:83
6FB0:13 23 13 23 13 23 13 23:0B
6FB8:13 23 13 23 13 23 13 23:13
6FC0:13 23 13 23 13 63 13 23:9B
6FC8:13 23 13 23 13 23 13 23:23
6FD0:13 23 13 23 13 23 13 23:2B
6FD8:13 23 13 23 13 23 13 23:33
6FE0:13 63 13 23 13 23 13 23:BB
6FE8:13 23 13 23 13 23 13 23:43
6FF0:13 23 13 23 13 23 13 23:4B
6FF8:13 23 13 23 13 E3 13 23:D3
```

```
8000:CD E2 8D CD B0 8C 01 0D:13
8008:BC ED 49 01 00 BD ED 49:2E
8010:01 0C BC ED 49 01 30 BD:3C
8018:ED 49 21 0A 95 CD EB 8C:26
8020:21 1B 95 CD EB 8C 21 35:58
8028:95 CD EB 8C 21 4F 95 CD:C2
8030:EB 8C 21 6A 95 CD EB 8C:42
8038:21 7D 95 CD EB 8C 21 97:44
8040:95 CD EB 8C 21 A9 95 CD:F6
8048:EB 8C CD 6A 8E 3E 40 CD:3C
8050:48 8E C2 63 80 3E 41 CD:29
8058:48 8E C2 7C 80 CD A3 8E:A5
8060:C3 4A 80 3E 48 32 F0 8E:43
8068:3C 32 F1 8E 3C 32 EE 8E:DD
8070:3C 32 EF 8E 00 3C 32 F2:EB
8078:8E C3 92 80 AF 32 F0 8E:59
8080:3C 32 EF 8E 3C 32 F1 8E:04
8088:3E 08 32 EE 8E 3E 2F 32:B7
8090:F2 8E CD B0 8C 21 07 8F:70
8098:11 08 8F 01 AF 04 36 00:E7
80A0:ED B0 AF 32 EC 8E 32 ED:F8
80A8:8E 32 F6 8E 32 F3 8E 32:52
80B0:F9 8E 32 02 8F 32 FE 8E:3C
80B8:32 FF 8E 32 00 8F 32 01:9A
80C0:8F 32 FA 8E 32 FB 8E 32:A7
80C8:FC 8E 32 FD 8E 32 05 8F:77
80D0:32 EA 8E 32 EB 8E 3E 01:6D
80D8:32 F8 8E 3E 30 32 03 8F:C5
80E0:21 00 28 22 D8 8E 21 80:F4
80E8:42 22 D4 8E 22 D6 8E CD:9A
80F0:6B 88 21 D8 95 CD 07 8D:FE
80F8:21 FB 95 CD 07 8D CD 4D:F6
8100:8D CD 76 8D CD 6A 8E CD:05
8108:A3 8E 3E 42 CD 48 8E C2:AD
8110:03 80 3A F3 8E B7 C2 81:74
8118:86 3A EE 8E CD 48 8E C4:48
8120:F2 82 3A EF 8E CD 48 8E:FD
8128:C4 01 83 3A F0 8E CD 48:BF
8130:8E C4 35 83 3A F1 8E CD:84
8138:48 8E C4 12 83 3A F2 8E:AA
8140:CD 48 8E 28 10 3A 06 8F:B6
8148:B7 20 0E 3E 01 32 02 8F:99
8150:32 06 8F 18 04 AF 32 06:D8
8158:8F 2A EC 8E 23 22 EC 8E:17
8160:7C FE 30 CA 66 86 3A F5:71
8168:8E B7 28 04 3D 32 F5 8E:ED
8170:CD A4 88 2A D4 8E 11 00:35
8178:20 01 07 07 CD 71 82 CD:E9
8180:33 8B CD 38 89 CD 3A 82:B2
8188:3A 02 8F B7 C4 D8 89 3A:43
8190:FD 8E EE 01 32 FD 8E 28:02
8198:0D 3A 01 8F B7 28 07 3D:75
81A0:32 01 8F C4 71 84 3A FC:D5
81A8:8E EE 01 32 FC 8E 20 0D:E6
81B0:3A 00 8F B7 28 07 3D 32:21
81B8:00 8F C4 22 84 3A FB 8E:68
81C0:EE 01 32 FB 8E 20 0D 3A:64
81C8:FF 8E B7 28 07 3D 32 FF:10
81D0:8E C4 D3 83 3A FA 8E EE:3C
81D8:01 32 FA 8E 28 0D 3A FE:80
81E0:8E B7 28 07 3D 32 FE 8E:B0
81E8:C4 D3 83 06 F5 ED 78 1F:0B
81F0:30 FB 3A 03 8F EE 20 32:20
81F8:03 8F 01 0D BC ED 49 01:12
8200:00 BD ED 49 01 0C BC ED:C0
8208:49 04 ED 79 21 90 01 2B:EA
8210:7D B4 20 FB 2A D6 8E 01:83
8218:07 07 CD C3 82 CD FD 88:A5
8220:CD 83 88 2A D4 8E 22 D6:CB
8228:8E 3A F6 8E B7 C4 0F 86:8A
8230:AF 32 F6 8E 32 02 8F C3:E6
8238:04 81 3A 03 8F 07 07 1E:90
8240:C0 2E C1 67 EE 80 57 CD:2C
8248:59 82 3A 03 8F 07 07 1E:F7
8250:80 2E 81 F6 04 67 EE 80:09
8258:57 06 08 C5 E5 D5 01 3F:DF
8260:00 ED B0 D1 3E 08 82 57:BC
8268:E1 3E 08 84 67 C1 10 EB:C0
8270:C9 79 32 82 82 3A 03 8F:BE
8278:07 07 EE 80 B4 67 C5 E5:52
8280:D5 06 00 1A B7 28 15 E6:91
8288:AA 28 06 4F 7E E6 55 B1:07
8290:77 1A E6 55 28 06 4F 7E:C8
8298:E6 AA B1 77 23 7D E6 3F:22
82A0:28 03 13 10 DE D1 3A 82:AB
82A8:82 6F 26 00 19 EB E1 7C:02
82B0:C6 08 67 E6 38 20 08 7D:CD
82B8:C6 40 6F 7C CE C0 67 C1:1C
82C0:10 BC C9 79 32 D3 82 3A:53
82C8:03 8F 07 07 EE 80 B4 67:66
82D0:C5 E5 01 00 00 5D 54 13:F6
82D8:36 00 0B ED B0 E1 7C C6:AF
82E0:08 67 E6 38 20 08 7D C6:35
82E8:40 6F 7C CE C0 67 C1 10:25
82F0:DF C9 2A D4 8E 7D E6 3F:A7
82F8:C8 2D 22 D4 8E CD A3 8E:CB
8300:C9 2A D4 8E 7D E6 3F FE:D2
8308:34 C8 2C 22 D4 8E CD A3:86
8310:8E C9 2A D4 8E 7D E6 C0:7F
8318:FE 40 20 06 7C E6 07 FE:82
8320:04 C8 CD DB 8C CD DB 8C:F1
8328:CD DB 8C CD DB 8C 22 D4:23
8330:8E CD A3 8E C9 2A D4 8E:21
8338:7D E6 C0 20 04 7C FE 41:BA
8340:C8 CD CB 8C CD CB 8C CD:B5
8348:CB 8C CD CB 8C 22 D4 8E:05
8350:CD A3 8E C9 CD A3 8E E6:99
8358:03 CA 69 83 FE 01 CA 6F:B3
8360:83 FE 02 CA 75 83 C3 7B:18
8368:83 3E 08 32 FE 8E C9 3E:83
8370:08 32 FF 8E C9 3E 08 32:BD
8378:00 8F C9 3E 08 32 01 8F:3F
8380:C9 CD B7 89 78 B7 C8 CD:31
8388:A3 8E E6 0F FE 0E 30 F7:0A
8390:B7 21 3F 41 28 07 01 40:C6
8398:00 09 3D 20 FC DD 75 01:B9
83A0:DD 74 02 DD 75 07 DD 74:80
83A8:08 11 47 24 DD 73 03 DD:BA
83B0:72 04 01 03 08 DD 71 05:51
83B8:DD 70 06 AF DD 77 0A DD:0F
83C0:77 0D 3E 01 DD 77 09 3E:DC
83C8:07 DD 77 00 3A F8 8E DD:2D
83D0:77 0C C9 CD B7 89 78 B7:3A
83D8:C8 CD A3 8E E6 0F FE 0E:18
83E0:30 F7 B7 21 3F 41 28 07:3B
83E8:01 40 00 09 3D 20 FC DD:CD
83F0:75 01 DD 74 02 DD 75 07:F4
83F8:DD 74 08 11 2F 24 DD 73:02
8400:03 DD 72 04 01 03 08 DD:DE
8408:71 05 DD 70 06 AF DD 77:59
8410:0A 3E 01 DD 77 09 3E 06:FC
8418:DD 77 00 3A F8 8E DD 77:3E
8420:0C C9 CD B7 89 78 B7 C8:43
8428:CD A3 8E E6 0F FE 0E 30:22
8430:F7 B7 21 3F 41 28 07 01:E6
8438:40 00 09 3D 20 FC DD 75:46
8440:01 DD 74 02 DD 75 07 DD:0B
8448:74 08 11 0F 24 DD 73 03:DE
8450:DD 72 04 01 04 08 DD 71:7C
8458:05 DD 70 06 AF DD 77 0A:2D
8460:3E 01 DD 77 09 3E 05 DD:43
8468:77 00 3A F8 8E DD 77 0C:86
8470:C9 CD B7 89 78 B7 C8 CD:22
8478:A3 8E E6 0F FE 0E 30 F7:FB
8480:B7 21 3F 41 28 07 01 40:B7
8488:00 09 3D 20 FC DD 75 01:AA
8490:DD 74 02 DD 75 07 DD 74:71
8498:08 11 6F 23 DD 73 03 DD:1F
84A0:72 04 01 04 08 DD 71 05:46
84A8:DD 70 06 AF DD 77 0A 3E:08
84B0:01 DD 77 09 3E 03 DD 77:83
84B8:00 3A F8 8E DD 77 0C C9:87
84C0:CD CD 85 DD 6E 01 DD 66:15
```

```
84C8:Ø2 7D E6 3F CA C8 85 2B:93
84DØ:7D E6 3F CA C8 85 2B 7D:9D
84D8:E6 3F CA C8 85 2B DD 7E:D4
84EØ:ØA B7 28 32 3D DD 77 ØA:ØC
84E8:DD 7E ØD B7 28 14 7D E6:23
84FØ:CØ FE CØ 2Ø Ø8 7C E6 47:82
84F8:FE 41 CA 65 87 CD DB 8C:BC
8648:DB 8C CD DB 8C CD 6B 87:43
865Ø:E1 23 23 23 CD 6B 87 DD:ØE
8658:E1 21 8Ø 42 22 D4 8E C9:55
866Ø:3E Ø1 32 F3 8E C9 21 E3:Ø3
8668:94 CD EB 8C 21 F6 94 CD:EA
867Ø:EB 8C Ø6 14 21 ØØ ØØ 2B:58
87CØ:DD 77 ØØ 3E Ø1 DD 77 Ø9:C6
87C8:DD 77 ØA DD 77 ØB C9 DD:A8
87DØ:7E ØB 3C E6 Ø1 DD 77 ØB:Ø3
87D8:C8 DD 7E ØA 3C FE Ø6 28:ØD
87EØ:14 DD 77 ØA DD 6E Ø3 DD:A4
87E8:66 Ø4 Ø1 ØC ØØ Ø9 DD 75:F9
87FØ:Ø3 DD 74 Ø4 C9 AF DD 77:6A
```




```
85ØØ:18 26 7D E6 CØ FE 4Ø 2Ø:6C
85Ø8:Ø8 7C E6 Ø7 FE Ø4 CA 65:17
851Ø:87 CD CB 8C 18 12 E5 CD:D6
8518:A3 8E E6 Ø1 DD 77 ØD CD:FØ
852Ø:A3 8E E1 E6 ØF DD 77 ØA:A9
8528:DD 75 Ø1 DD 74 Ø2 C9 CD:22
853Ø:CD 85 DD 6E Ø1 DD 66 Ø2:E8
8538:7D E6 3F CA C8 85 2B DD:Ø6
854Ø:7E ØA 3C FE 1Ø DD 77 ØA:12
8548:38 15 7D E6 3F CA C8 85:D5
855Ø:2B DD 7E ØA FE 1D 38 Ø7:CØ
8558:7D E6 3F CA C8 85 2B DD:26
856Ø:75 Ø1 DD 74 Ø2 C9 CD CD:86
8568:85 DD 6E Ø1 DD 66 Ø2 7D:25
857Ø:E6 3F 28 54 2B DD 7E ØA:E8
8578:3C E6 Ø3 DD 77 ØA 2Ø Ø5:19
858Ø:CD CB 8C 18 Ø7 FE Ø3 28:D8
8588:F7 CD DB 8C DD 75 Ø1 DD:5D
859Ø:74 Ø2 C9 CD CD 85 DD 6E:B6
8598:Ø1 DD 66 Ø2 7D E6 3F 28:EØ
85AØ:27 2B DD 75 Ø1 DD 74 Ø2:7C
85A8:DD 6E Ø3 DD 66 Ø4 Ø1 2Ø:8Ø
85BØ:ØØ Ø9 DD 7E ØA 3C FE Ø6:92
85B8:2Ø Ø4 AF 21 6F 23 DD 75:A6
85CØ:Ø3 DD 74 Ø4 DD 77 ØA C9:17
85C8:AF DD 77 ØØ C9 3A F5 8E:87
85DØ:B7 CØ DD 6E Ø1 DD 66 Ø2:E8
85D8:22 E4 8E CD 4C 88 ED 53:E4
85EØ:E2 8E 2A D4 8E CD 4C 88:19
85E8:2A E2 8E 7D DD 86 Ø5 4F:Ø9
85FØ:7B B9 DØ 7B C6 Ø7 4F 7D:D7
85F8:B9 DØ 7C DD 86 Ø6 4F 7A:79
86ØØ:B9 DØ 7A C6 Ø7 4F 7C B9:8E
86Ø8:DØ 3E Ø1 32 F6 8E C9 3A:16
861Ø:F4 8E B7 28 4B 3D 32 F4:4E
8618:8E 21 ØC 96 3E Ø2 1E Ø1:Ø6
862Ø:CD AE 8D 21 ØC 96 CD Ø7:8D
8628:8D 3E 28 32 F5 8E 2A D4:DA
863Ø:8E DD E5 E5 E5 E5 CD 6B:ØB
8638:87 E1 23 CD CB 8C CD CB:D6
864Ø:8C CD 6B 87 E1 23 23 CD:DD
8678:7D B4 2Ø FB 1Ø F6 C3 Ø3:BØ
868Ø:8Ø 21 C5 95 CD EB 8C Ø6:F2
8688:Ø6 21 ØØ ØØ 2B 7D B4 2Ø:Ø7
869Ø:FB 1Ø F6 C3 Ø3 8Ø DD 6E:A9
8698:Ø1 DD 66 Ø2 22 E4 8E DD:DF
86AØ:22 E8 8E CD 4C 88 ED 53:B5
86A8:E2 8E DD 21 Ø7 8F Ø6 64:ØE
86BØ:C5 DD 7E ØØ FE Ø3 2Ø Ø5:3F
86B8:CD DD 86 18 17 FE Ø5 2Ø:41
86CØ:Ø5 CD DD 86 18 ØE FE Ø6:82
86C8:2Ø Ø5 CD DD 86 18 Ø5 FE:94
86DØ:Ø7 CC DD 86 Ø1 ØF ØØ DD:EB
86D8:Ø9 C1 1Ø D4 C9 DD 6E Ø1:8E
86EØ:DD 66 Ø2 CD 4C 88 2A E2:2B
86E8:8E 7D C6 Ø2 4F 7B B9 DØ:4C
86FØ:7B DD 86 Ø5 4F 7D B9 DØ:59
86F8:7C C6 Ø4 4F 7A B9 DØ 7A:Ø6
87ØØ:DD 86 Ø6 4F 7C B9 DØ DD:18
87Ø8:22 E6 8E CD FA 87 Ø1 FF:66
871Ø:FF DD E5 CD 97 87 DD E1:63
8718:Ø1 FE FF CD 97 87 CD B7:3D
872Ø:89 2A E4 8E CD DB 8C CD:C7
8728:DB 8C CD DB 8C CD DB 8C:5C
873Ø:CD AØ 87 DD 2A E8 8E DD:D9
8738:46 ØC DD 2A E6 8E DD 7E:29
874Ø:ØC 4F B8 38 Ø3 98 18 Ø2:DØ
8748:3E ØØ DD 77 ØC B7 2Ø Ø4:Ø6
875Ø:AF DD 77 ØØ DD 2A E8 8E:BA
8758:DD 7E ØC 99 DD 77 ØC B7:67
876Ø:CØ DD 77 ØØ C9 DD 6E Ø1:EB
8768:DD 66 Ø2 E5 Ø1 FF FF Ø9:72
877Ø:E5 CD B7 89 E1 CD AØ 87:8A
8778:CD B7 89 E1 E5 Ø1 FE FF:C2
878Ø:Ø9 CD AØ 87 CD B7 89 E1:B8
8788:CD DB 8C CD DB 8C CD DB:6C
879Ø:8C CD DB 8C C3 AØ 87 C5:66
8798:CD B7 89 C1 2A E4 8E Ø9:4D
87AØ:DD 75 Ø1 DD 74 Ø2 DD 75:68
87A8:Ø7 DD 74 Ø8 11 5F 24 DD:DF
87BØ:73 Ø3 DD 72 Ø4 Ø1 Ø2 Ø6:67
87B8:DD 71 Ø5 DD 7Ø Ø6 3E Ø4:A7
87F8:ØØ C9 DD 7E ØØ FE Ø3 2Ø:A9
88ØØ:1Ø 21 Ø3 96 1E 78 3E Ø6:83
88Ø8:CD C8 8D 21 Ø3 96 C3 Ø7:38
881Ø:8D FE Ø5 2Ø 1Ø 21 Ø3 96:8B
8818:1E 96 3E Ø6 CD C8 8D 21:5Ø
882Ø:Ø3 96 C3 Ø7 8D FE Ø6 2Ø:1B
8828:1Ø 21 Ø3 96 1E C8 3E Ø6:8B
883Ø:CD C8 8D 21 Ø3 96 C3 Ø7:6Ø
8838:8D FE Ø7 CØ 21 Ø3 96 1E:1F
884Ø:FF 3E Ø6 CD C8 8D 21 Ø3:BE
8848:96 C3 Ø7 8D 7D E6 3F 5F:BB
885Ø:7C E6 Ø7 87 87 87 87 87:Ø7
8858:87 57 7C E6 38 ØF ØF ØF:74
886Ø:82 57 7D E6 CØ ØF ØF ØF:22
8868:82 57 C9 DD 21 5E 96 Ø6:12
887Ø:1D C5 CD A3 8E E6 55 DD:F7
8878:77 Ø3 Ø1 Ø6 ØØ DD Ø9 C1:ØD
888Ø:1Ø EF C9 DD 21 5E 96 Ø6:E8
8888:1D C5 DD 6E Ø4 DD 66 Ø5:3A
889Ø:3A Ø3 8F Ø7 Ø7 EE 8Ø B4:F8
8898:67 36 ØØ Ø1 Ø6 ØØ DD Ø9:68
88AØ:C1 1Ø E6 C9 DD 21 5E 96:38
88A8:Ø6 1D C5 DD 6E ØØ DD 66:94
88BØ:Ø1 DD 75 Ø4 DD 74 Ø5 DD:76
88B8:7E Ø2 DD 46 Ø3 FE Ø2 28:C2
88CØ:11 78 E6 AA 2Ø Ø7 CB 1Ø:7A
88C8:DD 7Ø Ø3 18 1D CB 18 DD:59
88DØ:7Ø Ø3 7D E6 3F 28 Ø9 2B:9F
88D8:DD 75 ØØ DD 74 Ø1 18 ØA:DD
88EØ:7D F6 3F 6F DD 75 ØØ DD:41
88E8:74 Ø1 3A Ø3 8F Ø7 Ø7 EE:36
88FØ:8Ø B4 67 7Ø Ø1 Ø6 ØØ DD:66
88F8:Ø9 C1 1Ø AE C9 Ø6 64 DD:A8
89ØØ:21 Ø7 8F C5 DD 7E Ø9 B7:B5
89Ø8:28 25 DD 7E ØØ B7 2Ø Ø4:DC
891Ø:AF DD 77 Ø9 DD 6E Ø7 DD:89
8918:66 Ø8 DD 4E Ø5 DD 46 Ø6:47
892Ø:CD C3 82 DD 6E Ø1 DD 75:D5
8928:Ø7 DD 66 Ø2 DD 74 Ø8 Ø1:F5
893Ø:ØF ØØ DD Ø9 C1 1Ø CC C9:84
8938:Ø6 64 DD 21 Ø7 8F C5 DD:72
```

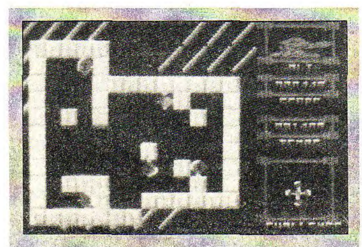
```
8940:7E ØØ B7 28 3F FE Ø1 2Ø:42
8948:Ø5 CD D3 8A 18 36 FE Ø2:CF
8950:2Ø Ø5 CD 64 8C 18 2D FE:71
8958:Ø3 2Ø Ø5 CD 93 85 18 24:2C
8960:FE Ø4 2Ø Ø5 CD CF 87 18:AF
8968:1B FE Ø5 2Ø Ø5 CD 66 85:7Ø
8970:18 12 FE Ø6 2Ø Ø5 CD 2F:18
8978:85 18 Ø9 FE Ø7 2Ø Ø5 CD:37
8980:CØ 84 18 ØØ Ø1 ØF ØØ DD:6Ø
8988:Ø9 C1 1Ø B2 Ø6 64 DD 21:1D
8990:Ø7 8F C5 DD 7E ØØ B7 28:B8
8998:15 DD 6E Ø1 DD 66 Ø2 DD:E9
89A0:5E Ø3 DD 56 Ø4 DD 4E Ø5:Ø8
89A8:DD 46 Ø6 CD 71 82 Ø1 ØF:AØ
89B0:ØØ DD Ø9 C1 1Ø DC C9 Ø6:39
89B8:64 DD 21 Ø7 8F DD 7E Ø9:91
89C0:B7 C8 11 ØF ØØ DD 19 1Ø:5C
89C8:F4 C9 3E 3Ø 32 13 96 32:67
89D0:14 96 21 11 96 C3 Ø7 8D:59
89D8:3A F9 8E FE Ø4 38 Ø2 C6:D1
89E0:FD 5F 47 3A F7 8E B7 28:49
89E8:E1 3D 32 F7 8E 1Ø F4 21:18
89F0:11 96 3E Ø2 CD AE 8D 21:7Ø
89F8:11 96 CD Ø7 8D CD A3 8E:B5
8A00:3A F9 8E FE Ø1 CA 21 8A:5Ø
8A08:FE Ø2 CA 25 8A FE Ø3 CA:91
8A10:2C 8A FE Ø4 CA 21 8A FE:52
8A18:Ø5 CA 25 8A FE Ø6 CA 2C:D2
8A20:8A CD 36 8A C9 CD 45 8A:86
8A28:CD 54 8A C9 CD 36 8A CD:5C
8A30:45 8A CD 54 8A C9 2A D4:F8
8A38:8E Ø1 Ø2 ØØ C5 Ø1 Ø6 ØØ:61
8A40:Ø9 E5 C3 81 8A 2A D4 8E:D4
8A48:Ø1 Ø2 Ø2 C5 Ø1 Ø6 ØØ Ø9:62
8A50:E5 C3 81 8A 2A D4 8E Ø1:A4
8A58:Ø2 82 C5 Ø1 Ø6 ØØ Ø9 E5:CØ
8A60:C3 81 8A 2A D4 8E Ø1 Ø2:84
8A68:Ø4 C5 Ø1 Ø6 ØØ Ø9 E5 C3:2C
8A70:81 8A 2A D4 8E Ø1 Ø2 84:57
8A78:C5 Ø1 Ø6 ØØ Ø9 E5 C3 81:C3
8A80:8A CD B7 89 78 B7 2Ø Ø3:11
8A88:E1 C1 C9 E1 CD DB 8C DD:33
8A90:75 Ø1 DD 74 Ø2 DD 75 Ø7:9B
8A98:DD 74 Ø8 3A F9 8E FE Ø4:2A
8AA0:38 ØA 11 3B 2Ø Ø1 Ø2 Ø4:69
8AA8:3E Ø2 18 Ø8 11 31 2Ø Ø1:3F
8AB0:Ø2 Ø5 3E Ø1 DD 73 Ø3 DD:Ø4
8AB8:72 Ø4 DD 71 Ø5 DD 7Ø Ø6:9E
8AC0:DD 77 ØC C1 DD 71 ØA DD:62
8AC8:7Ø ØB 3E Ø1 DD 77 ØØ DD:99
8AD0:77 Ø9 C9 DD 6E Ø1 DD 66:19
8AD8:Ø2 7D E6 3F FE 3C 3Ø 4E:2A
8AE0:7D E6 CØ FE CØ 2Ø Ø5 7C:6E
8AE8:E6 Ø7 28 42 7D E6 CØ FE:EB
8AF0:8Ø 2Ø Ø7 7C E6 Ø7 FE Ø4:F9
8AF8:28 34 DD 7E ØA 85 6F DD:E2
8B00:7E ØB 47 B7 28 ØE CB 78:39
8B08:28 Ø7 CB B8 CD 22 8B 18:64
8B10:Ø3 CD 28 8B DD 75 Ø1 DD:DA
8B18:74 Ø2 DD E5 CD 96 86 DD:5D
8B20:E1 C9 CD DB 8C 1Ø FB C9:32
8B28:CD CB 8C 1Ø FB C9 AF DD:48
8B30:77 ØØ C9 3A Ø5 8F 3C E6:BC
8B38:Ø7 32 Ø5 8F FE Ø1 2Ø 46:85
8B40:2A D8 8E Ø1 Ø4 ØØ Ø9 22:B6
8B48:D8 8E 5E 23 56 CB 7A CB:8B
8B50:BA D5 E5 Ø6 ØØ C4 26 8C:Ø8
8B58:E1 D1 CB 72 CB B2 D5 E5:AD
8B60:C4 54 83 E1 D1 23 ED 53:5E
8B68:DA 8E ED 53 DE 8E 5E 23:5Ø
8B70:56 CB 7A CB BA D5 E5 Ø6:94
8B78:Ø1 C4 26 8C E1 D1 ED 53:8C
8B80:DC 8E ED 53 EØ 8E 3A Ø5:88
8B88:8F 3D E6 Ø7 FE Ø7 ED 5B:5D
8B90:DA 8E 2Ø ØA ED 5B DE 8E:DE
8B98:2A DA 8E 22 DE 8E 83 5F:6A
8BA0:7A CE ØØ 57 21 FF 78 3A:54
8BA8:Ø3 8F Ø7 Ø7 EE 8Ø B4 67:4F
8BB0:EB CD FB 8B 3A Ø5 8F 3D:EE
8BB8:E6 Ø7 FE Ø7 ED 5B DC 8E:84
8BC0:2Ø ØA ED 5B EØ 8E 2A DC:6C
8BC8:8E 22 EØ 8E 83 5F 7A CE:8C
8BD0:ØØ 57 21 BF 44 3A Ø3 8F:A5
8BD8:Ø7 Ø7 EE 8Ø B4 67 EB CD:Ø5
8BE0:E3 8B C9 3E Ø8 ED AØ 1B:A5
8BE8:23 23 23 23 23 23 23 14:E7
8BF0:14 14 14 14 14 14 14 3D:93
8BF8:2Ø EB C9 3E Ø8 ED AØ 1B:BA
8C00:23 23 23 23 23 23 23 15:Ø8
8C08:15 15 15 15 15 15 15 3D:C8
8C10:2Ø EB C9 3E 63 32 F7 8E:31
8C18:3E 39 32 13 96 32 14 96:8C
8C20:21 11 96 C3 Ø7 8D C5 CD:F9
8C28:B7 89 78 C1 B7 C8 DD 7Ø:B7
8C30:ØE 3E Ø2 DD 77 ØØ 3E Ø1:CD
8C38:DD 77 Ø9 21 3F 42 DD 75:A8
8C40:Ø1 DD 74 Ø2 DD 75 Ø7 DD:13
8C48:74 Ø8 78 B7 11 43 2Ø 28:A3
8C50:Ø3 11 8B 2Ø DD 73 Ø3 DD:22
8C58:72 Ø4 Ø1 Ø6 ØC DD 71 Ø5:22
8C60:DD 7Ø Ø6 C9 DD 6E Ø1 DD:B3
8C68:66 Ø2 7D E6 3F 28 3C 2B:94
8C70:DD 75 Ø1 DD 74 Ø2 CD 4C:85
8C78:88 D5 2A D4 8E CD 4C 88:EC
8C80:E1 7D C6 Ø7 4F 7B B9 DØ:51
8C88:7B C6 Ø6 4F 7D B9 DØ 7C:CØ
8C90:C6 Ø7 4F 7A B9 DØ 7A C6:C8
8C98:ØC 4F 7C B9 DØ DD 7E ØE:46
8CA0:B7 28 Ø5 CD 13 8C 18 Ø3:DD
8CA8:CD 76 8D AF DD 77 ØØ C9:B3
8CB0:21 ØØ 4Ø Ø1 FE 3F 11 Ø1:1Ø
8CB8:4Ø 36 ØØ ED BØ 21 ØØ CØ:DA
8CC0:Ø1 FE 3F 11 Ø1 CØ 36 ØØ:49
8CC8:ED BØ C9 7C C6 F8 67 E6:9B
8CD0:4Ø CØ 7D C6 CØ 6F 7C CE:D9
8CD8:3F 67 C9 7C C6 Ø8 67 E6:CB
8CE0:38 CØ 7D C6 4Ø 6F 7C CE:61
8CE8:CØ 67 C9 E5 23 23 ØE ØØ:D4
8CF0:7E B7 28 Ø4 ØC 23 18 F8:66
8CF8:79 B7 E1 C8 5E 23 56 23:48
8D00:3E 2Ø 91 B3 5F 18 Ø4 5E:Ø1
8D08:23 56 23 7E B7 C8 FE Ø1:Ø2
8D10:23 CA Ø7 8D 2B D5 E5 D6:65
8D18:2Ø 6F 26 ØØ 29 29 29 29:3F
8D20:Ø1 ØØ 1Ø Ø9 CB FA D5 E5:DØ
8D28:CD 39 8D E1 D1 CB BA CD:74
8D30:39 8D E1 23 D1 13 13 18:ØB
8D38:D2 3E Ø8 ED AØ ED AØ 1B:C5
8D40:1B 14 14 14 14 14 14 14:A4
8D48:14 3D 2Ø EF C9 21 Ø5 96:Ø5
8D50:Ø1 Ø5 ØØ 11 Ø6 96 3E 3Ø:ØØ
8D58:77 ED BØ 32 ØE 96 3E 33:22
8D60:32 ØF 96 3E 39 32 13 96:75
8D68:32 14 96 3E 63 32 F7 8E:55
8D70:3E Ø3 32 F4 8E C9 3A F9:81
8D78:8E FE Ø6 C8 3C 32 F9 8E:58
8D80:FE Ø1 21 16 96 CA Ø7 8D:ØB
8D88:FE Ø2 21 1F 96 CA Ø7 8D:39
8D90:FE Ø3 21 28 96 CA Ø7 8D:67
8D98:FE Ø4 21 31 96 CA Ø7 8D:95
8DA0:FE Ø5 21 4Ø 96 CA Ø7 8D:DB
8DA8:21 4F 96 C3 Ø7 8D 4F Ø6:8C
8DB0:ØØ Ø9 23 23 41 E5 2B 7E:Ø4
8DB8:3D 77 FE 2F 2Ø Ø5 3E 39:5E
8DC0:77 1Ø F3 1D E1 2Ø ED C9:19
8DC8:4F Ø6 ØØ Ø9 23 23 41 E5:44
8DD0:2B 7E 3C 77 FE 3A 2Ø Ø5:6E
8DD8:3E 3Ø 77 1Ø F3 1D E1 2Ø:3C
8DE0:ED C9 F3 21 47 8E 22 39:B6
8DE8:ØØ Ø1 Ø1 BC ED 49 Ø1 2Ø:C8
8DF0:BD ED 49 Ø1 Ø2 BC ED 49:28
8DF8:Ø1 2B BD ED 49 Ø1 Ø6 BC:44
8E00:ED 49 Ø1 19 BD ED 49 Ø1:BA
8E08:Ø7 BC ED 49 Ø1 1D BD ED:46
8E10:49 Ø1 ØD BC ED 49 Ø1 ØØ:5E
8E18:BD ED 49 Ø1 ØC BC ED 49:83
8E20:Ø1 3Ø BD ED 49 Ø6 7F ØE:74
8E28:1Ø ED 49 ØE 54 ED 49 21:EC
8E30:BB 8E ØE ØØ 16 1Ø ED 49:5Ø
8E38:7E 23 F6 4Ø ED 79 ØC 15:DF
8E40:2Ø F4 Ø1 8C 7F ED 49 C9:59
8E48:21 B1 8E F5 E6 F8 ØF ØF:Ø6
8E50:ØF 5F 16 ØØ 19 F1 E5 21:5B
8E58:62 8E E6 Ø7 5F 19 7E E1:1D
8E60:A6 C9 Ø1 Ø2 Ø4 Ø8 1Ø 2Ø:E5
8E68:4Ø 8Ø 21 B1 8E Ø1 ØE F4:2B
8E70:ED 49 Ø6 F6 ED 78 E6 3Ø:A2
8E78:4F F6 CØ ED 79 ED 49 Ø4:3F
8E80:3E 92 ED 79 C5 CB F1 Ø6:7D
8E88:F6 ED 49 Ø6 F4 ED 78 2F:EB
8E90:77 23 ØC 79 E6 ØF FE ØA:FD
8E98:2Ø ED C1 3E 82 ED 79 Ø5:EA
8EA0:ED 49 C9 Ø6 Ø8 21 CB 8E:ØB
8EA8:4E 7E 81 77 4F 23 1Ø F9:74
8EB0:C9 ØØ ØØ ØØ ØØ ØØ ØØ ØØ:Ø7
8EB8:ØØ ØØ ØØ 14 ØB ØA ØE ØC:CB
8EC0:1C Ø4 15 1F 13 16 12 Ø2:9E
8EC8:ØØ 14 18 7B 2A Ø8 4A F3:52
8ED0:ØD 59 A7 2Ø ØØ ØØ ØØ ØØ:92
```

```
94E0:ØØ ØØ ØØ 8Ø C2 43 4F 4E:69
94E8:47 52 41 54 55 4C 41 54:52
94F0:49 4F 4E 2Ø 21 ØØ ØØ C3:92
94F8:5A 45 47 4F 4E 2Ø 49 53:5E
9500:2Ø 4E 4F 57 2Ø 46 52 45:44
9508:45 ØØ ØØ CØ 44 45 41 54:3B
9510:48 27 53 2Ø 54 49 43 4B:3B
9518:45 54 ØØ 8Ø C1 31 2Ø 2D:CD
9520:2Ø 53 54 41 52 54 2Ø 57:A5
9528:49 54 48 2Ø 4A 4F 59 53:59
9530:54 49 43 4B ØØ ØØ C2 32:7E
9538:2Ø 2D 2Ø 53 54 41 52 54:FB
9540:2Ø 57 49 54 48 2Ø 4B 45:2B
9548:59 42 4F 41 52 44 ØØ ØØ:DD
9550:C4 44 45 41 54 48 27 53:Ø1
9558:2Ø 54 49 43 4B 45 54 2Ø:FD
9560:43 4F 5Ø 59 52 49 47 48:AB
9568:54 ØØ 4Ø C4 52 2E 4F 2E:68
9570:53 2E 2Ø 53 4F 46 54 2Ø:DB
9578:31 39 39 3Ø ØØ 8Ø C4 5Ø:F7
9580:52 4F 47 52 41 4D 2Ø 41:1D
9588:4E 44 2Ø 47 52 41 5Ø 48:FF
9590:49 43 53 2Ø 42 59 ØØ CØ:CD
9598:C4 52 4F 59 45 52 2Ø 53:A3
95A0:45 42 41 53 54 49 45 4E:BA
95A8:ØØ ØØ C5 45 53 5Ø 45 43:1A
95B0:49 41 4C 59 2Ø 46 4F 52:55
95B8:2Ø 4D 49 43 52 4F 2Ø 2D:AA
95C0:2Ø 4D 41 47 ØØ 8Ø 42 47:F4
95C8:2Ø 41 2Ø 4D 2Ø 45 2Ø 2Ø:B1
95D0:4F 2Ø 56 2Ø 45 2Ø 52 ØØ:CD
95D8:CØ 44 53 43 4F 52 45 3A:E4
95E0:2Ø 3Ø 3Ø 3Ø 3Ø 3Ø 3Ø 2Ø:A5
95E8:4C 49 46 45 3A 2Ø 3Ø 33:ØB
95F0:2Ø 43 41 47 49 4F 3A 2Ø:E7
95F8:39 39 ØØ 15 45 54 59 5Ø:CC
9600:45 3A ØØ CE 44 3Ø 3Ø 3Ø:CB
9608:3Ø 3Ø 3Ø Ø1 E8 44 3Ø 33:CA
9610:Ø1 FC 44 39 39 ØØ 21 45:7B
9618:53 49 4D 5Ø 4C 45 ØØ 21:DC
9620:45 44 4F 55 42 4C 45 ØØ:B9
9628:21 45 54 52 49 5Ø 4C 45:36
9630:ØØ 21 45 53 55 5Ø 45 52:1F
9638:2Ø 53 49 4D 5Ø 4C 45 ØØ:DE
9640:21 45 53 55 5Ø 45 52 2Ø:3A
9648:44 4F 55 42 4C 45 ØØ 21:E9
9650:45 53 55 5Ø 45 52 2Ø 54:D5
9658:52 49 5Ø 4C 45 ØØ 6E 42:5D
9660:Ø2 41 6E 42 9E 42 Ø2 41:84
9668:9E 42 ØF 44 Ø2 41 ØF 44:76
9670:3E 41 Ø2 41 3E 41 22 43:92
9678:Ø1 54 22 43 47 42 Ø2 41:2E
9680:47 42 A4 43 Ø2 41 A4 43:FD
9688:69 43 Ø1 54 69 43 17 44:CØ
9690:Ø2 41 17 44 FC 42 Ø1 54:1E
9698:FC 42 AØ 43 Ø1 54 AØ 43:ØF
96A0:C9 42 Ø2 41 C9 42 55 41:61
96A8:Ø1 54 55 41 B1 42 Ø1 54:92
96B0:B1 42 58 43 Ø2 41 58 43:9F
96B8:38 41 Ø1 54 38 41 68 43:E9
96C0:Ø1 54 68 43 1B 42 Ø2 41:6C
96C8:1B 42 DB 43 Ø2 41 DB 43:3F
96D0:45 42 Ø2 41 45 42 16 44:D8
96D8:Ø1 54 16 44 57 41 Ø2 41:B8
96E0:57 41 1C 44 Ø1 54 1C 44:94
96E8:6B 42 Ø2 41 6B 42 48 44:32
96F0:Ø1 54 48 44 6Ø 44 Ø1 54:36
96F8:6Ø 44 42 43 Ø2 41 42 43:BE
9700:68 42 Ø2 41 68 42 54 42:7D
9708:Ø2 41 54 42 39 43 Ø2 41:EC
9710:39 43 B6 41 Ø1 54 B6 41:8B
```

Physique amusante

LA MOLÉCULE

N'ayez crainte, nul logiciel éducatif, mais un redoutable casse-tête ayant pour décor les confins de la matière.



Une configuration moléculaire vous est proposée par l'ordinateur. A vous de reconstituer le modèle par l'assemblage de divers atomes au moyen d'un téléporteur. Les indications nécessaires sont incluses.

Sauvegarde

Sauvez sous le nom évident de «MOLECULE», le programme Basic principal. Entrez ensuite par Amsaisie V.2 en vous reportant à son mode d'emploi, les codes hexadécimaux du second listing. Spécifiez 9700 comme adresse de début et sauvegardez le fichier binaire sous le nom «MOLBIN». Si vous le désirez, morcelez votre travail en plusieurs fichiers (M1, M2, etc.). Ces derniers devront ultérieurement être chargés à la suite (LOAD "M1" : LOAD "M2", etc.) après un MEMORY &96FF et sauvegardés ainsi dans un fichier unique:
SAVE "MOLBIN",b, &9700, &9AD

Claude Le Moullec

```
1Ø REM  :::::::::::::::::::::::     [1736]
2Ø REM  :                     :     [419]
3Ø REM  :     LA MOLECULE     :     [1ØØ2]
4Ø REM  :                     :     [419]
5Ø REM  :::::::::::::::::::::::     [1736]
6Ø REM  :                     :     [419]
7Ø REM  :     LMC SOFTWARE    :     [538]
8Ø REM  :                     :     [419]
9Ø REM  :  Claude LE MOULLEC  :     [2Ø31]
1ØØ REM :                     :     [419]
11Ø REM :::::::::::::::::::::::     [1736]
12Ø REM                             [272]
13Ø REM :::::::::::::::::::::::     [1736]
14Ø REM :                     :     [419]
15Ø REM :     REDEFINITION    :     [1622]
16Ø REM :                     :     [419]
17Ø REM :::::::::::::::::::::::     [1736]
18Ø SYMBOL AFTER 169                [1435]
19Ø SYMBOL 17Ø,17Ø,17Ø,17Ø,17Ø,17   [2823]
Ø,17Ø,17Ø,17Ø
2ØØ SYMBOL 171,Ø,84,84,84,84,84,8   [1597]
4,Ø
21Ø SYMBOL 2Ø1,Ø,Ø,Ø,Ø,1,1,Ø,1      [1667]
22Ø SYMBOL 2Ø2,14,31,123,245,226,   [29Ø5]
255,127,1Ø3
23Ø SYMBOL 2Ø3,128,224,192,224,25   [322Ø]
2,255,255,224
24Ø SYMBOL 2Ø4,Ø,Ø,Ø,58,239,255,2   [2ØØ1]
54,188
25Ø SYMBOL 2Ø5,3,3,1,Ø,1,3,15,62    [1814]
26Ø SYMBOL 2Ø6,1Ø8,113,191,22Ø,17   [2683]
9,119,127,248
27Ø SYMBOL 2Ø7,15,252,128,Ø,192,2   [253Ø]
24,253,195
28Ø SYMBOL 2Ø8,24Ø,Ø,Ø,Ø,Ø,Ø,Ø,12   [1947]
8
29Ø SYMBOL 2Ø9,124,249,Ø,Ø,Ø,Ø,Ø,   [1555]
Ø
3ØØ SYMBOL 21Ø,231,223,Ø,Ø,Ø,Ø,Ø,   [2Ø16]
Ø
31Ø SYMBOL 211,63,255,Ø,Ø,Ø,Ø,Ø,Ø   [2345]
32Ø SYMBOL 212,192,192,Ø,Ø,Ø,Ø,Ø,   [1836]
Ø
33Ø SYMBOL 213,254,188,94,47,255,   [2394]
255,126,192
34Ø SYMBOL 214,254,252,3Ø,15,255,   [2281]
193,Ø,Ø
35Ø REM :::::::::::::::::::::::     [1736]
36Ø REM :                     :     [419]
37Ø REM :  VARIABLES DE BASE  :     [2Ø81]
38Ø REM :                     :     [419]
39Ø REM :::::::::::::::::::::::     [1736]
4ØØ MEMORY &96FF:LOAD "|MOLBIN",&   [1714]
97ØØ
41Ø GOSUB 363Ø                      [1194]
42Ø DEFINT a-z:CALL &BBFF:MODE Ø:   [1865]
BORDER Ø
43Ø RESTORE 44Ø:FOR h=Ø TO 15:REA   [294Ø]
D a:INK h,a:NEXT
44Ø DATA Ø,1,15,5,3,22,14,7,16,9,   [1985]
```

```
13,24,18,23,26,6
450 INK 7,7,18:tr$=CHR$(22)+CHR$(  [3976]
1):nr$=CHR$(22)+CHR$(Ø)
460 DEF FN lt(x,y)=&CØØØ+(y-1)*8Ø  [1711]
+(x-1)*2
470 ali3$=CHR$(2Ø9)+CHR$(21Ø)+CHR  [2257]
$(211)+CHR$(212)
48Ø ali2$=CHR$(2Ø5)+CHR$(2Ø6)+CHR  [2118]
$(2Ø7)+CHR$(2Ø8)
49Ø ali1$=CHR$(2Ø1)+CHR$(2Ø2)+CHR  [3744]
$(2Ø3)+CHR$(2Ø4)
5ØØ DIM sp(2Ø):FOR H=Ø TO 15:SP(H  [4722]
+1)=&97ØØ+(H*64):NEXT:SP(2Ø)=&A4Ø
Ø
51Ø DIM JE(16,13):WINDOW #2,17,19  [2387]
,18,23
52Ø SC=Ø:POKE &AØ39,2ØØ:TTT=21Ø:T  [1467]
A=1
53Ø ON BREAK GOSUB 267Ø:REM ligne  [2858]
 a n'entrer que quand prog au poi
nt
54Ø GOTO 72Ø                       [421]
55Ø REM ::::::::::::::::::::::     [1736]
56Ø REM :                    :     [419]
57Ø REM :    DECOR DE BASE   :     [1019]
58Ø REM :                    :     [419]
59Ø REM ::::::::::::::::::::::     [1736]
6ØØ FOR H=Ø TO 3Ø:CALL &AØ32,&CØØ  [315Ø]
Ø+H:CALL &AØ32,&CØ3B-H:NEXT
61Ø DEF FN po(x,y)=&CØØØ+(y-1)*24  [2654]
Ø+(x-1)*6:POKE &AØ1A,24:POKE &AØ1
E,6
62Ø RESTORE 66Ø:FOR g=1 TO 4:READ  [3312]
 a$:FOR h=1 TO 1Ø
63Ø b$=MID$(a$,h,1):a=VAL(b$):IF   [2963]
a=Ø THEN 65Ø
64Ø CALL &AØØD,FN po(h,g),&9BAØ    [1829]
65Ø NEXT h,g                       [382]
66Ø DATA 1111ØØØ11111,111ØØØ11111  [2523]
1,11ØØØØØØØ11,1ØØØØØØØØ1Ø
67Ø RESTORE 71Ø:FOR g=5 TO 8:READ  [231Ø]
 a$:FOR h=1 TO 1Ø
68Ø b$=MID$(a$,h,1):a=VAL(b$):IF   [3Ø16]
a=Ø THEN 7ØØ
69Ø CALL &AØØD,FN po(h,g)+8Ø,&9BA  [239Ø]
Ø
7ØØ NEXT h,g:POKE &AØ1A,16:POKE &  [118Ø]
AØ1E,4:RETURN
71Ø DATA ØØØØØØ1ØØØ,ØØØØ111ØØØ,ØØ  [2737]
Ø1111ØØØ,ØØ11111ØØØ
72Ø PLOT -1Ø,-1Ø,11:TAG:MOVE 498,  [5Ø78]
368:PRINT ali1$;:MOVE 498,352:PRI
NT ali2$;:MOVE 498,336:PRINT ali3
$;:TAGOFF
73Ø DATA 16,1,5,16,5,5,16,7,5,16,  [3Ø41]
9,5,16,12,5,16,14,5,16,17,5,16,24
,5
74Ø DATA 16,1,5,2Ø,1,5,16,7,3,2Ø,  [3716]
7,3,16,12,3,2Ø,12,3,16,17,8,2Ø,17
,8
75Ø e1=12:e2=9:RESTORE 73Ø:FOR i=  [5745]
1 TO 16:GOSUB 76Ø:NEXT:GOTO 84Ø
76Ø READ X,Y,L:x1=14+(X-1)*32:y1=  [2428]
388-(Y-1)*16
77Ø IF i>8 THEN 81Ø                [1ØØ7]
78Ø FOR J=Ø TO 2 STEP 2:PLOT x1,y  [2455]
1+j,e1:DRAW x1+(l-1)*32,y1+j:NEXT
79Ø FOR J=4 TO 6 STEP 2:PLOT x1,y  [4175]
1+j,e2:DRAW x1+(l-1)*32,y1+j:NEXT
8ØØ RETURN                         [555]
81Ø PLOT x1+4,y1+2,e1:DRAW x1+4,y  [2465]
1+2-(l-1)*16
82Ø PLOT x1,y1+2,e2:DRAW x1,y1+2-  [1Ø96]
(l-1)*16
83Ø RETURN                         [555]
84Ø LOCATE 1,1:PRINT tr$:RESTORE   [4834]
85Ø:FOR h=1 TO 16:GOSUB 87Ø:NEXT:
GOTO 89Ø
85Ø DATA 16,1,16,5,2Ø,1,2Ø,5,16,7  [3113]
,2Ø,7,16,9,2Ø,9,16,12,2Ø,12
86Ø DATA 16,14,2Ø,14,16,17,2Ø,17,  [1479]
16,24,2Ø,24
87Ø READ x,y:PEN 12:LOCATE x,y:PR  [352Ø]
INT CHR$(17Ø):PEN 9:LOCATE x,y
88Ø PRINT CHR$(171):RETURN         [1611]
89Ø RESTORE 9ØØ:FOR h=1 TO 5:READ  [3477]
 ZL,YL,PL,a$:GOSUB 91Ø:NEXT:GOTO
94Ø
9ØØ DATA 32,8,Ø,"ØØØØØØ",32,1Ø,1,  [3729]
"SCORE",32,13,Ø,"ØØØØØØ",32,15,1,
"TEMPS",31,25,Ø,"CHALLENGE"
91Ø A$=UPPER$(A$):FOR T=1 TO LEN(  [27Ø6]
A$):SP=(ASC(MID$(A$,T,1)))-48
92Ø IF sp<Ø THEN sp=43             [1Ø71]
93Ø CALL &9DØØ,FN LT(ZL,YL)+(T*2)  [4127]
+PL,&9D2Ø+(SP*16):NEXT:RETURN
94Ø A$="   ALI":FOR T=1 TO LEN(A$  [3598]
):SP=(ASC(MID$(A$,T,1)))-48
95Ø IF sp<Ø THEN sp=43             [1Ø71]
96Ø CALL &9DØØ,&C1CB+(T*2),&9D2Ø+  [2869]
(SP*16):NEXT:GOSUB 349Ø
97Ø REM  ::::::::::::::::::::::    [1736]
98Ø REM  :                    :    [419]
99Ø REM  :   DESSIN TABLEAU   :    [1952]
1ØØØ REM :                    :    [419]
1Ø1Ø REM ::::::::::::::::::::::    [1736]
1Ø2Ø GOSUB 6ØØ:ON TA GOTO 1Ø3Ø,1Ø  [5Ø84]
4Ø,1Ø5Ø,1Ø6Ø,1Ø7Ø,1Ø8Ø,1Ø9Ø,11ØØ,
111Ø,112Ø,113Ø,114Ø,115Ø
1Ø3Ø RESTORE 273Ø:GOSUB 117Ø:GOTO  [2417]
 141Ø
1Ø4Ø RESTORE 279Ø:GOSUB 117Ø:GOTO  [2779]
 141Ø
1Ø5Ø RESTORE 285Ø:GOSUB 117Ø:GOTO  [1923]
 141Ø
1Ø6Ø RESTORE 291Ø:GOSUB 117Ø:GOTO  [2767]
 141Ø
1Ø7Ø RESTORE 297Ø:GOSUB 117Ø:GOTO  [1923]
 141Ø
1Ø8Ø RESTORE 3Ø3Ø:GOSUB 117Ø:GOTO  [2Ø63]
 141Ø
1Ø9Ø RESTORE 3Ø9Ø:GOSUB 117Ø:GOTO  [2672]
 141Ø
11ØØ RESTORE 315Ø:GOSUB 117Ø:GOTO  [2Ø63]
 141Ø
111Ø RESTORE 321Ø:GOSUB 117Ø:GOTO  [1666]
 141Ø
112Ø RESTORE 327Ø:GOSUB 117Ø:GOTO  [2Ø63]
 141Ø
113Ø RESTORE 333Ø:GOSUB 117Ø:GOTO  [1229]
 141Ø
114Ø RESTORE 339Ø:GOSUB 117Ø:GOTO  [2ØØ5]
```

```
 1410
1150 TA=1:GOTO 1020                     [743]
1160 REM ::: DESSIN TABLEAU :::         [1859]
1170 POKE &A020,0:DEF FN po(x,y)=       [1592]
&E000+(y-1)*160+(x-1)*4
1180 ERASE JE:DIM JE(16,13):FOR H       [3702]
=1 TO 16:JE(H,1)=20:JE(H,13)=20:N
EXT
1190 FOR H=2 TO 12:JE(1,H)=20:JE(       [2212]
16,H)=20:NEXT:
1200 FOR g=1 TO 12:READ a$:FOR h=       [2105]
1 TO 15
1210 b$=MID$(a$,h,1):IF B$="X" TH       [4255]
EN JE(H,G)=20:GOTO 1270
1220 IF B$="G" THEN A=16:GOTO 124       [1216]
0
1230 a=VAL("&"+B$):IF A=0 THEN 12       [2065]
60
1240 IF A=7 OR A=12 THEN mx=h:my=       [921]
g
1250 CALL &A00D,FN PO(H,G),SP(A):       [2548]
JE(H,G)=A:GOTO 1270
1260 CALL &A00D,FN PO(H,G),&9C50:       [2282]
JE(H,G)=0
1270 NEXT H,G                           [382]
1280 READ X,Y:POKE &A020,&AE:CALL       [2858]
 &A00D,FN PO(X,Y),SP(5)
1290 DEF FN po(x,y)=&C000+(y-1)*8       [2532]
0+(x-1)*2:POKE &A01A,8:POKE &A01E
,2
1300 CLS #2:FOR G=19 TO 23:READ A       [2199]
$:FOR H=1 TO 5
1310 b$=MID$(a$,h,1):a=VAL("&"+b$       [2900]
):IF A=0 THEN 1330
1320 CALL &A00D,FN PO(H+33,G)-1,&       [2640]
9AF0+(A*16)
1330 NEXT H,G:POKE &A01A,16:POKE        [1020]
&A01E,4
1340 LOCATE 1,1:PRINT NR$:TTT=TTT       [1492]
-10:tien=0
1350 fin=0:GAN=0:AF=5:READ tts,MU       [3432]
L:REC=TTT:GOSUB 2410:RETURN
1360 REM ::::::::::::::::::::::::       [1736]
1370 REM :                      :       [419]
1380 REM :  ROUTINE PRINCIPALE  :       [2225]
1390 REM :                      :       [419]
1400 REM ::::::::::::::::::::::::       [1736]
1410 POKE &A020,&AE:DEF FN po(x,y       [2120]
)=&E000+(y-1)*160+(x-1)*4
1420 EVERY 250,0 GOSUB 2470             [492]
1430 SON=0:IF FIN=1 THEN 2540           [1869]
1440 IF INKEY(0)*INKEY(72)=0 THEN       [2029]
 1510
1450 IF INKEY(2)*INKEY(73)=0 THEN       [943]
 1640
1460 IF INKEY(8)*INKEY(74)=0 THEN       [1330]
 1770
1470 IF INKEY(1)*INKEY(75)=0 THEN       [838]
 1890
1480 IF INKEY(9)*INKEY(76)=0 THEN       [1257]
 2050
1490 GOTO 1430                          [341]
1500 REM ::: VERS LE HAUT :::           [881]
1510 IF je(x,y-1)>19 THEN 1430          [594]
1520 IF TIEN=1 THEN 1580                [1413]
1530 GOSUB 1540:y=y-1:GOTO 1430         [1985]
1540 CALL &A00D,FN po(x,y),sp(AF)       [2057]
:CALL &A052,FN po(x,y)
1550 FOR I=1 TO 3:CALL &A05C,sp(A       [1279]
F)
1560 FOR t=1 TO 20:NEXT t:CALL &A       [2294]
06E,sp(AF):NEXT I
1570 CALL &A05C,sp(AF):RETURN           [2223]
1580 IF je(x,y-1)<>0 THEN 1430          [820]
1590 JE(X,Y)=0:GOSUB 1540:Y=Y-1:J       [3354]
E(X,Y)=ACT
1600 SON=SON+1:SOUND 1,40+(SON*10       [2367]
),3,5,,,6
1610 IF act=7+MUL THEN mx=x:my=y:       [4667]
GOTO 1510 ELSE 1510
1620 REM ::: VERS LE BAS :::            [1275]
1630 GOTO 1510                          [397]
1640 IF je(x,y+1)>19 THEN 1430          [788]
1650 IF TIEN=1 THEN 1720                [1417]
1660 GOSUB 1670:y=y+1:GOTO 1430         [1301]
1670 CALL &A00D,FN po(x,y),sp(AF)       [2057]
:CALL &A052,FN po(x,y)
1680 FOR I=1 TO 3:CALL &A07B,sp(A       [1476]
F)
1690 FOR t=1 TO 20:NEXT t:CALL &A       [2294]
06E,sp(AF):NEXT I
1700 CALL &A07B,sp(AF):RETURN           [1288]
1710 REM ::::AVANCE bas                 [1603]
1720 IF je(x,y+1)<>0 THEN 1430          [559]
1730 JE(X,Y)=0:GOSUB 1670:Y=Y+1:J       [1182]
E(X,Y)=ACT
1740 SON=SON+1:SOUND 1,40+(SON*10       [2367]
),3,5,,,6
1750 IF act=7+MUL THEN mx=x:my=y:       [4045]
GOTO 1640 ELSE 1640
1760 REM ::: A GAUCHE :::               [1052]
1770 IF je(x-1,y)>19 THEN 1430          [598]
1780 IF TIEN=1 THEN 1840                [1409]
1790 GOSUB 1800:x=x-1:GOTO 1430         [2098]
1800 CALL &A00D,FN po(x,y),sp(AF)       [2596]
:add=FN po(x,y)
1810 FOR I=1 TO 3:CALL &A00D,add-       [1631]
I,sp(AF)
1820 FOR t=1 TO 20:NEXT t:CALL &A       [3227]
00D,add-I,sp(AF):NEXT I
1830 CALL &A00D,add-4,sp(AF):RETU       [1800]
RN
1840 IF je(x-1,y)<>0 THEN 1430          [818]
1850 JE(X,Y)=0:GOSUB 1800:X=X-1:J       [1853]
E(X,Y)=ACT
1860 SON=SON+1:SOUND 1,40+(SON*10       [2367]
),3,5,,,6
1870 IF act=7+MUL THEN mx=x:my=y:       [4182]
GOTO 1770 ELSE 1770
1880 REM ::: A DROITE :::               [1375]
1890 IF je(x+1,y)>19 THEN 1430          [1206]
1900 IF TIEN=1 THEN 1960                [1417]
1910 GOSUB 1920:x=x+1:GOTO 1430         [1801]
1920 CALL &A00D,FN po(x,y),sp(AF)       [2596]
:add=FN po(x,y)
1930 FOR I=1 TO 3:CALL &A00D,add+       [1370]
I,sp(AF)
1940 FOR t=1 TO 20:NEXT t:CALL &A       [2536]
00D,add+I,sp(AF):NEXT I
1950 CALL &A00D,add+4,sp(AF):RETU       [3125]
RN
1960 IF je(x+1,y)<>0 THEN 1430          [942]
```

```
1970 JE(X,Y)=Ø:GOSUB 192Ø:X=X+1:J [2639]
E(X,Y)=ACT
198Ø SON=SON+1:SOUND 1,4Ø+(SON*1Ø [2367]
),3,5,,,6
199Ø IF act=7+MUL THEN mx=x:my=y: [38Ø1]
GOTO 189Ø ELSE 189Ø
2ØØØ REM :::::::::::::::::::::::: [1736]
2Ø1Ø REM :                      : [419]
2Ø2Ø REM :    PRISE / LACHER    : [16Ø3]
2Ø3Ø REM :                      : [419]
2Ø4Ø REM :::::::::::::::::::::::: [1736]
2Ø5Ø IF JE(X,Y)<7 OR JE(X,Y)>17 T [185Ø]
HEN 143Ø
2Ø6Ø IF TIEN=1 THEN 213Ø [1415]
2Ø7Ø ENT 3,1ØØ,-5Ø,3Ø:SOUND 1,6ØØ [14Ø6]
,1Ø,6,Ø,3
2Ø8Ø CALL &AØØD,FN PO(X,Y),SP(AF) [3Ø68]
:CALL &AØØD,FN PO(X,Y),SP(6)
2Ø9Ø CALL &AØ9Ø,FN PO(X,Y):TIEN=1 [2454]
21ØØ ACT=JE(X,Y):WHILE INKEY$<>"" [1419]
:WEND
211Ø FOR T=1 TO 5ØØ:NEXT:AF=2Ø:GO [2413]
TO 143Ø
212Ø REM ::: LACHER ::: [317]
213Ø ENT 3,1ØØ,-5Ø,3Ø:SOUND 1,6ØØ [14Ø6]
,1Ø,6,Ø,3
214Ø CALL &AØØD,FN PO(X,Y),SP(6): [2675]
CALL &AØØD,FN PO(X,Y),SP(5)
215Ø TIEN=Ø:WHILE INKEY$<>"":WEND [1159]
216Ø GOSUB 222Ø:AF=5:GOTO 143Ø [1518]
217Ø REM :::::::::::::::::::::::: [1736]
218Ø REM :                      : [419]
219Ø REM : MOLECULE CORRECTE ? : [13Ø4]
22ØØ REM :                      : [419]
221Ø REM :::::::::::::::::::::::: [1736]
222Ø DATA -1,Ø,-1,-1,Ø,-1,1,-1,1, [2Ø82]
Ø,1,1,Ø,1,-1,1
223Ø DATA -2,-1,-2,Ø,-2,1,2,-1,2, [226Ø]
Ø,2,1
224Ø tout=Ø:RESTORE 222Ø:FOR h=1 [1933]
TO 14:READ px,py
225Ø px=px+mx:py=py+my:sm=je(px,p [25Ø1]
y)
226Ø IF sm<7 OR SM>16 THEN tot=Ø [32Ø2]
ELSE tot=sm*h
227Ø tout=tout+tot:NEXT:TOUT=TOUT [2594]
+JE(MX,MY)
228Ø IF TOUT=TTS THEN FIN=1:GAN=1 [2Ø58]
229Ø RETURN [555]
23ØØ REM :::::::::::::::::::::::: [1736]
231Ø REM :                      : [419]
232Ø REM :  GESTION COMPTEURS   : [2235]
233Ø REM :                      : [419]
234Ø REM :::::::::::::::::::::::: [1736]
235Ø IF sc=Ø THEN a$="ØØØØØØ":ZL= [2425]
32:YL=8:GOSUB 91Ø:RETURN
236Ø a$=STR$(sc):A$=RIGHT$(A$,LEN [2868]
(A$)-1)
237Ø IF sc<1Ø THEN ZL=36:YL=8:GOS [2766]
UB 91Ø:RETURN
238Ø IF sc<1ØØ THEN ZL=35:YL=8:GO [3486]
SUB 91Ø:RETURN
239Ø IF sc<1ØØØ THEN ZL=34:YL=8:G [2136]
OSUB 91Ø:RETURN
24ØØ ZL=33:YL=8:GOSUB 91Ø:RETURN [912]
241Ø IF REC=Ø THEN a$="ØØØØØØ":ZL [174Ø]
=32:YL=13:GOSUB 91Ø:RETURN
242Ø a$=STR$(REC):A$=RIGHT$(A$,LE [28Ø6]
N(A$)-1)
243Ø IF REC<1Ø THEN ZL=33:YL=13:a [23Ø7]
$="ØØØ"+a$:GOSUB 91Ø:RETURN
244Ø IF REC<1ØØ THEN ZL=33:YL=13: [4483]
a$="ØØ"+a$:GOSUB 91Ø:RETURN
245Ø IF REC<1ØØØ THEN ZL=33:YL=13 [1853]
:a$="Ø"+a$:GOSUB 91Ø:RETURN
246Ø REM :::: TEMPO [448]
247Ø DI:REC=REC-5:GOSUB 241Ø [845]
248Ø IF REC<5 THEN FIN=1:EI:RETUR [3633]
N ELSE EI:RETURN
249Ø REM :::::::::::::::::::::::: [1736]
25ØØ REM :                      : [419]
251Ø REM :     GAGNE/PERDU      : [1324]
252Ø REM :                      : [419]
253Ø REM :::::::::::::::::::::::: [1736]
254Ø FOR H=Ø TO 3:MU=REMAIN(H):NE [2Ø96]
XT
255Ø IF gan=Ø THEN 262Ø [638]
256Ø FOR H=1 TO 16:BORDER h:SOUND [3415]
 1,h*15,1Ø,14:FOR t=1 TO 2ØØ
257Ø NEXT t,h:BORDER Ø [244Ø]
258Ø ENV 1Ø,15,-1,1:FOR H=1 TO IN [3773]
T(REC/5):sc=sc+1:GOSUB 235Ø
259Ø SOUND 1,Ø,15,15,1Ø,,15:NEXT: [3944]
REM CLAUDE TEL 96 38 94 24
26ØØ TA=TA+1:REC=Ø:GOSUB 241Ø:GOT [1738]
O 1Ø2Ø
261Ø REM ::: PERDU ::: [1Ø27]
262Ø BORDER 26:INK Ø,26:SOUND 4,1 [22Ø3]
5ØØ,5Ø,7,Ø,Ø,1Ø
263Ø FOR t=1 TO 4ØØØ:NEXT t:BORDE [2617]
R Ø:INK Ø,Ø
264Ø SOUND 1,239,2Ø,6:SOUND 1,Ø,2 [7574]
,6:SOUND 1,239,2Ø,6:SOUND 1,319,2
Ø,6:SOUND 1,213,2Ø,6:SOUND 1,239,
4Ø,6:SOUND 1,319,2Ø,6
265Ø PEN 11:FOR H=1 TO 5:LOCATE 1 [7462]
8,3:PRINT CHR$(214):FOR T=1 TO 4Ø
Ø:NEXT T:LOCATE 18,3:PRINT CHR$(2
13):FOR T=1 TO 2ØØ:NEXT T,H
266Ø FOR T=1 TO 2ØØØ:NEXT T:TA=TA [3646]
-1:ttt=ttt+1Ø:GOTO 26ØØ
267Ø fin=1:gan=Ø:RETURN [96Ø]
268Ø REM :::::::::::::::::::::::: [1736]
269Ø REM :                      : [419]
27ØØ REM :  DATA 12 TABLEAUX   : [1172]
271Ø REM :                      : [419]
272Ø REM :::::::::::::::::::::::: [1736]
273Ø DATA XXXXXXXXXXXXXXX,XX11111 [4344]
XXXXXXXX,XX1ØØØ111111XXX
274Ø DATA XX1Ø92ØØØØØ1XXX,XX1Ø2ØØ [3Ø31]
Ø7ØØ1XXX,XX1Ø2ØØØØ33Ø1XX
275Ø DATA XX1ØØØØ2Ø3ØØ1XX,XX1ØØØØ [272Ø]
2Ø3ØØ1XX,XX113B3ØØØØØ1XX
276Ø DATA XXX1ØØØØØØØØ1XX,XXX1111 [3916]
111111XX,XXXXXXXXXXXXXXX
277Ø DATA 6,6,ØØØØØ,Ø513Ø,ØØØØØ,Ø [18Ø3]
ØØØØ,ØØØØØ,63,Ø
278Ø REM ::: TAB 2 ::: [329]
279Ø DATA XXXXXXXXXXXXXXX,X44444X [4198]
XXXXXXXX,X4ØØØ4XXX44444X
28ØØ DATA X4ØØB33333ØØ84X,X4ØØ2ØØ [2591]
ØØØØØ34X,X4Ø2ØØØ37ØØØØ4X
```

```
2810 DATA X4000000000004X,X400000 [3479]
0030004X,X4900033333334X
2820 DATA X111111XXXXXXX,XXXXXXX [3880]
XXXXXXX,XXXXXXXXXXXXXX
2830 DATA 8,6,00000,00200,05130,0 [1457]
0000,00000,87,0
2840 REM ::: TAB 3 :::            [623]
2850 DATA XXXXXXXXXXXXXX,X22222X [4496]
XXXXXXX,X300B3XXXXXXXX
2860 DATA X3000322222222X,X300030 [2619]
0071003X,X3010300000103X
2870 DATA X3000000000003X,X300000 [3048]
0100003X,X3000000A01803X
2880 DATA X3011100000103X,X300910 [3037]
0222222X,X2222222XXXXXX
2890 DATA 6,7,00000,00200,05130,0 [1996]
0400,00000,157,0
2900 REM ::: TAB 4 :::            [569]
2910 DATA XXXXXXXXXXXXXX,XXXXXXX [3521]
X323232X,X32323XX200003X
2920 DATA X20003XX300C02X,X30F023 [3574]
2300003X,X2000000000002X
2930 DATA X3000323230032,X204000 [2986]
00200003,X30G0000E3000D2
2940 DATA X2323232320032,XXXXXXX [3193]
XX323232,XXXXXXXXXXXXXX
2950 DATA 5,7,00000,0A070,00600,0 [1834]
9080,00000,300,5
2960 REM ::: TAB 5 :::            [545]
2970 DATA XXXXXXXXXXXXXX,X44444X [4652]
XXXXXXX,X40004XXXXXXXX
2980 DATA X400C4444444XXX,X400020 [2857]
0020444X,X4020000029004X
2990 DATA X4000000020004X,X400222 [3298]
0020204X,X4000000D00004X
3000 DATA X4000000200204X,X444444 [3001]
000B2F4X,XXXXXX44444444X
3010 DATA 5,7,00000,00073,00600,5 [2191]
9000,00000,413,5
3020 REM ::: TAB 6 :::            [679]
3030 DATA XXXXXXXXXXXXXX,XXX1212 [3415]
1212XXXX,XXX20000001XXXX
3040 DATA XXX1000B002XXXX,XXX2000 [2963]
33333333,XXX121000C4D002
3050 DATA X212B2000440001,X1F0010 [2603]
00000002,X20002003333333
3060 DATA X100000000E00C2,X2000G0 [2767]
00000001,X44444444444444
3070 DATA 7,7,00000,5A007,00660,5 [1860]
9008,00000,469,5
3080 REM ::: TAB 7 :::            [591]
3090 DATA XXXXXXXXXXXXXX,X111111 [3284]
111111XX,X17000090001XX
3100 DATA X100020220001XX,X102020 [1909]
000001XX,X1000000001111X
3110 DATA X111110A2080B1X,X108000 [2853]
0202001X,X1102200000071X
3120 DATA X1000000001111X,X170000 [3581]
02A1XXXX,X111111111XXXXX
3130 DATA 6,3,00000,02200,51113,0 [2155]
4400,00000,330,0
3140 REM ::: TAB 8 :::            [664]
3150 DATA XXXXXXXXXXXXXX,X333333 [4336]
33333XXX,X3000004F003XXX
3160 DATA X300400000C3XXX,X300404 [2589]
444033XX,X30G4040E4003XX
3170 DATA X304400004D03XX,X3C0000 [3476]
004003XX,X3004000000F3XX
3180 DATA X300440000443XX,X300G40 [3369]
000003XX,X333333333333XX
3190 DATA 6,3,00000,0AA07,00660,0 [1646]
9908,00000,729,5
3200 REM ::: TAB 9 :::            [640]
3210 DATA XXXXXXXXXXXXXX,X444444 [3453]
44444444,X4F03000000004
3220 DATA X4G03000000004,X4C0300 [1863]
00000004,X4C03000000004
3230 DATA X4703000000004,X480300 [2980]
00000004,X40000000000004
3240 DATA X40000000000004,X49A300 [2806]
00000004,X44444444444444
3250 DATA 6,3,00000,A0020,06613,9 [2370]
0040,00000,577,5
3260 REM ::: TAB 10 :::           [724]
3270 DATA XXXXXXXXXXXXXX,X333333 [3249]
33333333,XX400829002CB4X
3280 DATA XXX400022004XX,XXXX4A0 [3001]
02074XXX,XXXXX400004XXXX
3290 DATA XXXXXX4004XXXXX,XXXXXX4 [3184]
004XXXXX,XXXXX400004XXXX
3300 DATA XXXX400000A4XXX,XXX4800 [2809]
200074XX,X33333333333333
3310 DATA 4,3,00000,02020,51613,0 [2060]
4040,00000,469,5
3320 REM ::: TAB 11 :::           [732]
3330 DATA XXXXXXXXXXXXXX,XXXXXXX [4752]
XXXXXXXX,XXXX3444443XXXX
3340 DATA XXXX1C77881XXXX,XXXX1AA [2653]
9B01XXXX,XXXX1DEFG01XXXX
3350 DATA XXXX1010101XXXX,XXXX122 [3178]
2221XXXX,XXXX3444443XXXX
3360 DATA XXXXXXXXXXXXXX,XXXXXXX [4457]
XXXXXXXX,XXXXXXXXXXXXXX
3370 DATA 8,7,00000,2A072,13651,4 [1881]
9084,00000,943,5
3380 REM ::: TAB 12 :::           [772]
3390 DATA XXXXXXXXXXXXXX,X111111 [3263]
1111111X,X1200000000021X
3400 DATA X1000C77880001X,X1000AA [3023]
9B00001X,X10000DEFG0001X
3410 DATA X1000000000001X,X100000 [2716]
0000001X,X1000000000001X
3420 DATA X1000000000001X,X120000 [3088]
0000021X,X11111111111111X
3430 DATA 8,7,00000,2A072,13651,4 [1881]
9084,00000,943,5
3440 REM ::::::::::::::::::::::: [1736]
3450 REM :                     : [419]
3460 REM :    PRESENTATION     : [1647]
3470 REM :                     : [419]
3480 REM ::::::::::::::::::::::: [1736]
3490 a$="        LA MOLECULE":ZL= [2254]
1:YL=1:GOSUB 910
3500 a$="        :::::::::::":ZL= [3302]
1:YL=2:GOSUB 910
3510 a$="  EN VOUS SERVANT DES AT [3118]
OMES":ZL=1:YL=4:GOSUB 910
3520 a$="DISSEMINES DANS CHAQUE E [3004]
CRAN":ZL=1:YL=6:GOSUB 910
3530 a$="A  VOUS DE  RECONSTITUER [4605]
 LA":ZL=1:YL=8:GOSUB 910
3540 a$="MOLECULE DONT LE SCHEMA  [1945]
```

```
VOUS":ZL=1:YL=1Ø:GOSUB 91Ø
355Ø a$="EST FOURNI<":ZL=1:YL=12:  [2555]
GOSUB 91Ø
356Ø a$="  POUR CELA  VOUS UTILIS [1813]
EREZ":ZL=1:YL=14:GOSUB 91Ø
357Ø a$="LE TELEPORTEUR<":ZL=1:YL [31Ø2]
=16:GOSUB 91Ø
358Ø a$="  LA FIN D>UN DEPLACEMEN [3257]
T NE":ZL=1:YL=18:GOSUB 91Ø
359Ø a$="POUVANT SE FAIRE QU>APRE [3799]
S LA":ZL=1:YL=2Ø:GOSUB 91Ø
36ØØ a$="RENCONTRE D>UN OBSTACLE< [3182]
":ZL=1:YL=22:GOSUB 91Ø
361Ø a$="           BONNE CHANCE":ZL [2665]
=1:YL=24:GOSUB 91Ø
362Ø WHILE INKEY$="":WEND:MU=REMA [3116]
IN(1):RETURN
363Ø DIM b(62):RESTORE 364Ø:FOR n [3396]
=1 TO 62:READ b(n):NEXT
364Ø DATA 239,179,179,179,142,159 [11129]
,179,159,142,179,179,142,119,1Ø6,
1Ø6,1Ø6,119,142,179,159,179,159,1
42,179,213,213,239,179,179,1Ø6,11
9,142,142,179,159,179,159,1Ø6,119
,142,142,119,1Ø6,1Ø6,1Ø6,89,119,1
42,142,179,159,179,159,142,179,21
3,213,239,179,179,179,Ø
365Ø ENV 1,1,15,1,5,-3,2:ENV 2,15 [439Ø]
,-1,1Ø:ENT-1,1Ø,-1,1:ENT-2,1,-127
,1,5,24,1
366Ø EVERY 1Ø,1 GOSUB 387Ø         [663]
367Ø TR$=CHR$(23)+CHR$(1):NR$=CHR [2Ø28]
$(23)+CHR$(Ø)
368Ø INK Ø,Ø:INK 15,6:BORDER Ø:MO [1886]
DE Ø:FOR A=1 TO 14:INK A,26-A:NEX
T
369Ø FOR A=Ø TO PI STEP Ø.Ø5:X=SI [5554]
N(A)*4Ø:PLOT 32Ø+X,2ØØ+COS(A)*4Ø,
1:DRAWR -X*2,Ø:NEXT
37ØØ PLOT 47Ø,5Ø:P=2:FOR A=PI/2 T [5Ø22]
O 2.5*PI STEP Ø.Ø9:X=SIN(A)*15Ø:Y
=COS(A)*6Ø:DRAW 32Ø+X+Y,2ØØ+Y-X,P
:P=P+1+14*(P=15):GOSUB 374Ø:NEXT
371Ø P=2:PLOT 26Ø,26Ø:FOR A=Ø TO  [814Ø]
2*PI STEP Ø.Ø45:X=SIN(A)*15Ø:Y=CO
S(A)*6Ø:DRAW 32Ø-X-Y,2ØØ+Y-X,P:P=
P+1+14*(P=15):GOSUB 374Ø:NEXT
372Ø P=2:PLOT 39Ø,2ØØ:FOR A=Ø TO  [5Ø69]
2*PI STEP Ø.Ø225:X=SIN(A)*17Ø:Y=C
OS(A)*7Ø:DRAW 32Ø+Y,2ØØ+X,P:P=P+1
+14*(P=15):GOSUB 374Ø:NEXT:GOTO 3
75Ø
373Ø FOR A=2 TO 14:INK A,26:CALL  [3523]
&BD19:INK A,Ø:NEXT:GOTO 373Ø
374Ø IF p=15 THEN p=8:RETURN ELSE [2371]
 RETURN
375Ø ORIGIN Ø,Ø:INK 4,Ø:PLOT -1Ø, [2125]
-1Ø,4:a$="LA MOLECULE"
376Ø ENC=1:EN2=15:Y=33Ø:X=124:GOS [2275]
UB 379Ø
377Ø PEN 15:LOCATE 1Ø,19:PRINT TR [39Ø6]
$;"BY":a$="CLAUDE LE MOULLEC"
378Ø PLOT -1Ø,-1Ø,1:TAG:MOVE 46,4 [5Ø43]
Ø:PRINT A$;:PLOT -1Ø,-1Ø,15:MOVE
5Ø,36:PRINT A$;:TAGOFF:PRINT NR$:
GOTO 384Ø
379Ø TAG:FOR H=1 TO LEN(A$):B$=MI [31Ø4]
D$(A$,H,1):MOVE (H*36)-36,14:PRIN
T B$;:NEXT:TAGOFF
38ØØ FOR G=Ø TO 14 STEP 2:FOR H=Ø [1911]
 TO 432 STEP 2
381Ø IF TEST(H,2+G)=4 THEN PLOT H [3439]
+X,Y+(G*2),ENC:PLOT H+X,Y+2+(G*2)
,EN2
382Ø NEXT H,G:LOCATE 1,25:PRINT S [3898]
PACE$(18):INK 4,26:RETURN
383Ø PLOT H+X,Y+(G*2),ENC:PLOT H+ [1582]
X,Y+2+(G*2),EN2:RETURN
384Ø a=1                           [327]
385Ø a=a+1:IF a>14 THEN a=2       [1842]
386Ø INK A,26:CALL &BD19:INK A,Ø: [3783]
a$=INKEY$:IF a$="" THEN 385Ø ELSE
 RETURN
387Ø qq=qq+1:IF qq=62 THEN qq=1:w [2Ø97]
=1
388Ø IF qq=32 THEN w=3             [276]
389Ø SOUND 4,b(qq),19,14:w=w+1:IF [6656]
 w=5 THEN SOUND 2,Ø,1Ø,15,2,Ø,2:w
=1:RETURN ELSE SOUND 1,w*15Ø,6,15
,1,2,2:RETURN
39ØØ RETURN                        [555]
```

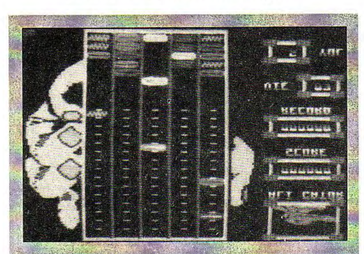
```
97ØØ:F3 F3 F3 F3 F3 F3 F3 E6:5B
97Ø8:F3 9C 79 CC F3 9C 79 CC:7B
971Ø:F3 9C 79 CC F3 9C 79 CC:83
9718:F3 9C 79 CC F3 9C 79 CC:8B
972Ø:F3 9C 79 CC F3 9C 79 CC:93
9728:F3 9C 79 CC F3 9C 79 CC:9B
973Ø:F3 9C 79 CC F3 9C 79 CC:A3
9738:F3 F3 F3 CC E6 CC CC CC:EB
974Ø:F3 F3 F3 F3 F3 F3 F3 E6:9B
9748:F3 F3 F3 CC F3 F3 F3 CC:37
975Ø:F3 9C 79 CC F3 9C 79 CC:C3
9758:F3 9C 79 CC F3 9C 79 CC:CB
976Ø:F3 9C 79 CC F3 9C 79 CC:D3
9768:F3 9C 79 CC F3 9C 79 CC:DB
977Ø:F3 F3 F3 CC F3 CC CC CC:64
9778:E6 CC CC CC CC CC CC CC:D9
978Ø:F3 F3 F3 F3 F3 F3 F3 E6:DB
9788:B7 3D 3D 6C B6 3E 3E 6E:DB
979Ø:B7 3D 3D 6C B6 3E 3E 6E:E3
9798:B7 3D 3D 6C B6 3E 3E 6E:EB
97AØ:B7 3D 3D 6C B6 3E 3E 6E:F3
97A8:B7 3D 3D 6C B6 3E 3E 6E:FB
97BØ:B7 3D 3D 6C B6 3E 3E 6E:Ø3
97B8:E6 CC CC CC CC CC CC CC:19
97CØ:F3 E6 F3 E6 B6 E6 B6 E6:A3
97C8:B6 E6 B6 E6 B6 E6 B6 E6:B7
97DØ:B6 E6 B6 E6 B6 E6 B6 E6:BF
97D8:F3 E6 F3 E6 CC CC CC CC:57
97EØ:F3 E6 F3 E6 B6 E6 B6 E6:C3
97E8:B6 E6 B6 E6 B6 E6 B6 E6:D7
97FØ:B6 E6 B6 E6 B6 E6 B6 E6:DF
97F8:F3 E6 F3 E6 CC CC CC CC:77
98ØØ:ØC ØC ØC ØC Ø8 ØØ ØØ Ø4:58
98Ø8:Ø8 ØØ ØØ Ø4 Ø8 ØØ ØØ Ø4:ØØ
981Ø:Ø8 ØØ ØØ Ø4 Ø8 ØØ ØØ Ø4:Ø8
9818:Ø8 ØØ ØØ Ø4 Ø8 ØØ ØØ Ø4:1Ø
982Ø:Ø8 ØØ ØØ Ø4 Ø8 ØØ ØØ Ø4:18
9828:Ø8 ØØ ØØ Ø4 Ø8 ØØ ØØ Ø4:2Ø
983Ø:Ø8 ØØ ØØ Ø4 Ø8 ØØ ØØ Ø4:28
9838:Ø8 ØØ ØØ Ø4 ØC ØC ØC ØC:2Ø
984Ø:ØØ 54 A8 ØØ ØØ A8 54 ØØ:B4
9848:54 ØØ ØØ A8 A8 ØØ ØØ 54:BC
985Ø:A8 ØØ ØØ 54 A8 ØØ ØØ 54:C8
9858:A8 ØØ ØØ 54 A8 ØØ ØØ 54:DØ
986Ø:A8 ØØ ØØ 54 A8 ØØ ØØ 54:D8
9868:A8 ØØ ØØ 54 A8 ØØ ØØ 54:EØ
987Ø:A8 ØØ ØØ 54 54 ØØ ØØ A8:E4
9878:ØØ A8 54 ØØ ØØ 54 A8 ØØ:EC
988Ø:ØØ 51 ØA ØØ ØØ 51 ØA ØØ:Ø4
9888:ØØ 41 82 ØØ ØØ C3 C3 ØØ:ØF
989Ø:ØØ C3 33 ØØ 41 93 33 22:73
9898:41 D6 76 22 4B D6 FC 73:FE
98AØ:E3 76 76 27 41 76 76 8A:98
98A8:41 33 67 8A ØØ 33 67 ØØ:47
98BØ:ØØ 33 CF ØØ ØØ 45 8A ØØ:7F
98B8:ØØ 51 ØA ØØ ØØ 51 ØA ØØ:3C
98CØ:ØØ 15 2A ØØ ØØ 3F ØC ØØ:CE
98C8:15 2E ØC Ø8 15 ØC ØC Ø8:5A
98DØ:2E 5D FF ØC 2E ØC AE ØC:CD
98D8:2E ØC AE ØC 2E ØC AE ØC:4Ø
98EØ:2E ØC AE ØC Ø4 5D FF Ø8:73
98E8:Ø4 ØC ØC Ø8 ØØ ØC ØC ØØ:7C
98FØ:ØØ ØC ØC ØØ ØØ Ø4 Ø8 ØØ:14
98F8:ØØ 51 ØA ØØ ØØ 51 ØA ØØ:7C
99ØØ:ØØ Ø4 F3 ØØ ØØ ØC 59 A2:41
99Ø8:ØØ ØC ØC A2 Ø4 ØC ØC F3:AD
991Ø:Ø4 ØC ØC 59 Ø4 5D FF 59:5Ø
9918:Ø4 ØC AE 59 2E ØC AE 59:F3
992Ø:ØE ØC AE 59 Ø4 ØC AE 59:33
9928:Ø4 5D FF 59 Ø4 ØC ØC 59:58
```

```
9930:00 0C 0C 08 00 0C 0C 08:01
9938:00 04 0C 00 00 04 0C 00:69
9940:00 15 0A 00 00 15 0A 00:E5
9948:00 04 08 00 00 0C 0C 00:9D
9950:00 0C 0C 00 04 0C 0C 08:15
9958:04 FF AE 08 0C 5D 0C 1D:C3
9960:0C 5D 0C 1D 0C 5D 0C 1D:FD
9968:0C 5D 0C 1D 0C FF AE 1D:3F
9970:04 0C 0C 2A 04 0C 1D 2A:68
9978:00 0C 3F 00 00 15 2A 00:8A
9980:00 F3 08 00 51 A6 0C 00:E4
9988:51 0C 0C 00 F3 0C 0C 08:49
9990:A6 0C 0C 08 A6 FF AE 08:65
9998:A6 5D 0C 08 A6 5D 0C 0D:FD
99A0:A6 5D 0C 59 A6 5D 0C 08:21
99A8:A6 FF AE 08 A6 0C 0C 08:29
99B0:04 0C 0C 00 04 0C 0C 08:39
99B8:00 0C 08 00 00 0C 08 00:01
99C0:2A 00 00 15 1F 00 00 2F:EA
99C8:05 0B 07 0A 00 03 FF 00:C4
99D0:01 57 FF 00 01 BA 30 AA:16
99D8:01 BA FF AA 01 BA FF AA:35
99E0:01 BA FF AA 01 BA FF AA:3D
99E8:01 BA 30 AA 01 57 FF 00:36
99F0:00 03 FF 00 05 0B 07 0A:68
99F8:1F 00 00 2F 2A 00 00 15:E6
9A00:00 00 00 00 00 51 F3 00:25
9A08:00 F3 CC 88 00 E6 CC 88:44
9A10:51 CC CC 88 51 D8 F0 CC:AC
9A18:51 CC E4 CC 51 CC E4 CC:70
9A20:51 CC E4 CC 51 CC E4 CC:78
9A28:51 D8 F0 CC 51 CC CC CC:14
9A30:00 E6 CC 88 05 E6 CC 88:6B
9A38:1F 51 E6 00 2A 00 00 00:17
9A40:2A 00 00 00 1F 51 E6 00:CF
9A48:05 E6 CC 88 00 E6 CC 88:6F
9A50:51 CC CC CC 51 D8 F0 CC:FC
9A58:51 CC E4 CC 51 CC E4 CC:B0
9A60:51 CC E4 CC 51 CC E4 CC:B8
9A68:51 D8 F0 CC 51 CC CC 88:34
9A70:00 E6 CC 88 00 F3 CC 88:E0
9A78:00 51 F3 00 00 00 00 00:8D
9A80:00 00 00 15 00 D9 A2 2F:6A
9A88:44 CC D9 0A 44 CC D9 00:BC
9A90:CC CC CC A2 CC F0 E4 A2:62
9A98:CC D8 CC A2 CC D8 CC A2:4A
9AA0:CC D8 CC A2 CC D8 CC A2:52
9AA8:CC F0 E4 A2 44 CC CC A2:E2
9AB0:44 CC D9 00 44 CC F3 00:72
9AB8:00 F3 A2 00 00 00 00 00:1E
9AC0:00 00 00 00 00 F3 A2 00:7A
9AC8:44 CC F3 00 44 CC D9 00:22
9AD0:44 CC CC A2 CC F0 E4 A2:1A
9AD8:CC D8 CC A2 CC D8 CC A2:8A
9AE0:CC D8 CC A2 CC D8 CC A2:92
9AE8:CC F0 E4 A2 CC CC CC A2:CA
9AF0:44 CC D9 00 44 CC D9 0A:4C
9AF8:00 D9 A2 2F 00 00 00 15:8E
9B00:05 A2 05 A2 93 63 1B 73:01
9B08:B3 27 93 63 51 0A 51 0A:41
9B10:51 A2 F3 0C A6 0C 0C 0C:83
9B18:0C 0C 0C 0C 05 A2 05 A2:43
9B20:04 A2 04 F3 86 59 A6 59:E1
9B28:0E 59 86 59 04 0C 04 08:31
9B30:51 0A 51 0A 0C 0C 0C 0C:83
9B38:0C 0C A6 0C F3 0C 51 A2:67
9B40:51 08 F3 08 A6 49 A6 0D:1B
9B48:A6 59 A6 49 0C 08 04 08:19
9B50:B2 71 1A 25 75 BA 75 BA:09
9B58:75 BA 75 BA 1A 25 B2 71:E9
9B60:00 88 44 CC 44 CC 44 CC:5F
9B68:44 CC 44 CC 0A 88 A2 00:AB
9B70:A2 00 0A 88 44 CC 44 CC:43
9B78:44 CC 44 CC 44 CC 00 88:47
9B80:00 51 44 05 CC 88 CC 88:9D
9B88:CC 88 CC 88 CC 88 44 00:8B
9B90:44 00 CC 88 CC 88 CC 88:F3
9B98:CC 88 CC 88 44 05 00 51:8D
9BA0:00 00 00 00 00 E7 00 00:A5
9BA8:00 00 00 E7 00 00 00 00:DF
9BB0:51 8E 00 00 00 00 51 8E:5F
9BB8:00 00 00 E7 5D 00 00 00:04
9BC0:00 00 E7 5D 00 00 00 51:0C
9BC8:8E AA 00 00 00 51 8E AA:5D
9BD0:00 00 00 E7 5D 00 00 00:D8
9BD8:00 E7 5D 00 00 00 51 8E:FF
9BE0:AA 00 00 00 51 8E AA 00:B4
9BE8:00 00 E7 5D 00 00 00 00:AC
9BF0:E7 5D 00 00 00 51 8E AA:44
9BF8:00 00 00 51 8E AA 00 00:99
9C00:00 E7 5D 00 00 00 00 E7:B9
9C08:5D 00 00 00 51 8E AA 00:90
9C10:00 00 51 8E AA 00 00 00:29
9C18:E7 5D 00 00 00 00 E7 5D:8E
9C20:00 00 00 00 8E AA 00 00:7E
9C28:00 00 8E AA 00 00 00 00:16
9C30:00 00 00 00 00 00 00 00:CC
9C38:00 00 00 00 00 00 00 00:D4
9C40:00 00 00 00 00 00 00 00:DC
9C48:00 00 00 00 00 00 00 00:E4
9C50:00 00 00 00 00 00 00 00:EC
9C58:00 00 00 00 00 00 00 00:F4
9C60:00 00 00 00 00 00 00 00:FC
9C68:00 00 00 00 00 00 00 00:04
9C70:00 00 00 00 00 00 00 00:0C
9C78:00 00 00 00 00 00 00 00:14
9C80:00 00 00 00 00 00 00 00:1C
9C88:00 00 00 00 00 00 00 00:24
9C90:00 00 00 00 00 00 00 00:2C
9C98:00 00 00 00 00 00 00 00:34
9CA0:00 00 00 00 00 00 00 00:3C
9CA8:00 00 00 00 00 00 00 00:44
9CB0:00 00 00 00 00 00 00 00:4C
9CB8:00 00 00 00 00 00 00 00:54
9CC0:00 00 00 00 00 00 00 00:5C
9CC8:00 00 00 00 00 00 00 00:64
9CD0:00 00 00 00 00 00 00 00:6C
9CD8:00 00 00 00 00 00 00 00:74
9CE0:00 00 00 00 00 00 00 00:7C
9CE8:00 00 00 00 00 00 00 00:84
9CF0:00 00 00 00 00 00 00 00:8C
9CF8:00 00 00 00 00 00 00 00:94
9D00:DD 5E 00 DD 56 01 DD 6E:D9
9D08:02 DD 66 03 06 08 C5 E5:78
9D10:06 02 1A 77 23 13 10 FA:42
9D18:E1 CD 26 BC C1 10 EF C9:88
9D20:15 3F 15 3F 45 45 45 45:8D
9D28:04 04 04 04 55 FF 00 00:90
9D30:00 2A 15 2A 00 8A 00 8A:94
9D38:00 08 00 08 55 FF 00 00:A8
9D40:15 3F 15 15 00 45 00 45:C9
9D48:00 08 04 00 55 FF 00 00:A4
9D50:15 3F 15 15 00 45 00 45:D9
9D58:00 0C 00 04 55 FF 00 00:C0
9D60:00 15 00 3F 00 CF 45 45:08
9D68:04 0C 00 04 00 55 00 00:2F
9D70:15 3F 15 00 45 00 00 CF:B0
9D78:00 04 04 04 55 FF 00 00:DC
9D80:15 3F 15 15 45 00 45 CF:F7
9D88:04 04 04 04 55 FF 00 00:F0
9D90:15 3F 00 15 00 45 00 8A:02
9D98:00 08 04 00 55 00 00 00:FA
9DA0:15 3F 15 15 45 45 45 CF:B5
9DA8:04 04 04 04 55 FF 00 00:10
9DB0:15 3F 15 15 45 45 45 CF:C5
9DB8:00 04 00 04 55 FF 00 00:10
9DC0:00 00 00 00 00 00 00 00:5D
9DC8:00 08 00 00 00 AA 00 00:71
9DD0:00 00 00 00 00 00 00 00:6D
9DD8:00 08 00 00 00 AA 55 00:D4
9DE0:00 00 00 00 00 00 00 00:7D
9DE8:00 00 00 08 00 AA 00 00:A1
9DF0:00 00 00 00 00 00 00 00:8D
9DF8:00 00 00 08 00 AA 55 00:04
9E00:00 2A 00 2A 45 00 00 00:F3
9E08:00 00 00 00 00 00 00 00:A6
9E10:00 2A 15 15 45 45 00 45:B4
9E18:00 08 00 00 00 AA 00 00:C2
9E20:00 2A 00 2A 00 8A 00 8A:46
9E28:00 08 00 00 00 AA 00 00:D2
9E30:15 3F 15 3F 45 45 45 45:9E
9E38:04 0C 04 04 55 55 00 00:B5
9E40:15 3F 15 3F 45 45 45 45:AE
9E48:04 08 04 04 55 FF 00 00:B9
9E50:15 3F 15 3F 45 00 45 00:F8
9E58:04 00 04 00 55 FF 00 00:A9
9E60:15 2A 15 3F 45 45 45 45:A4
9E68:04 04 04 04 55 FF 00 00:D1
9E70:15 3F 15 3F 45 00 45 00:18
9E78:04 08 04 00 55 FF 00 00:D9
9E80:15 3F 15 3F 45 00 45 00:28
9E88:04 08 04 00 55 00 00 00:EF
9E90:15 3F 15 3F 45 45 45 00:D6
9E98:04 00 04 04 55 FF 00 00:F9
9EA0:15 15 15 15 45 45 45 45:12
9EA8:04 0C 04 04 55 55 00 00:25
9EB0:15 3F 00 2A 00 8A 00 8A:15
9EB8:00 08 00 08 55 FF 00 00:29
9EC0:15 3F 00 2A 00 8A 00 8A:25
9EC8:00 08 00 08 55 AA 00 00:3B
9ED0:15 15 15 15 45 45 45 8A:6A
9ED8:04 08 04 04 55 55 00 00:4D
9EE0:15 00 15 00 45 00 45 00:0E
9EE8:04 00 04 00 55 FF 00 00:39
9EF0:15 15 15 3F 45 CF 45 45:46
9EF8:04 04 04 04 55 55 00 00:65
9F00:15 15 15 15 45 CF 45 CF:FF
9F08:04 0C 04 04 55 55 00 00:86
9F10:15 3F 15 15 45 45 45 45:D7
9F18:04 04 04 04 55 FF 00 00:82
9F20:15 3F 15 3F 45 45 45 45:8F
9F28:04 0C 04 00 55 00 00 00:98
9F30:15 3F 15 3F 45 45 45 45:9F
9F38:04 04 04 04 55 FF 00 AA:F2
9F40:15 3F 15 3F 45 45 45 45:AF
9F48:04 08 04 04 55 55 00 00:BE
9F50:15 3F 15 3F 45 00 45 00:F9
9F58:04 0C 00 04 55 FF 00 00:C6
9F60:15 3F 15 3F 00 8A 00 8A:59
9F68:00 08 00 08 00 AA 00 00:33
9F70:15 15 15 15 45 45 45 45:E3
9F78:04 04 04 04 55 FF 00 00:E2
9F80:15 15 15 15 45 45 45 45:F3
9F88:04 04 04 04 00 AA 00 00:4B
9F90:15 15 15 15 45 45 45 45:03
9F98:04 0C 04 0C 55 55 00 00:36
9FA0:15 15 15 15 45 45 00 8A:58
9FA8:00 08 04 04 55 55 00 00:1A
9FB0:15 15 15 15 45 45 00 8A:68
9FB8:00 08 00 08 00 AA 00 00:83
9FC0:15 3F 00 15 00 45 00 8A:34
9FC8:00 08 04 00 55 FF 00 00:26
9FD0:00 00 00 00 00 00 00 00:6F
9FD8:00 00 00 00 00 00 00 00:77
9FE0:00 00 00 00 00 00 00 00:7F
9FE8:00 00 00 00 00 00 00 00:87
9FF0:00 00 00 00 00 00 00 00:8F
9FF8:00 00 00 00 00 00 00 00:97
A000:7C C6 08 67 D0 D5 11 50:61
A008:C0 19 D1 C9 00 DD 5E 00:F1
A010:DD 56 01 DD 6E 02 DD 66:1D
A018:03 06 10 C5 E5 06 04 1A:94
A020:AE 77 13 23 10 F9 E1 CD:D6
A028:00 A0 C1 10 EE C9 00 00:E7
A030:00 00 DD 6E 00 DD 66 01:1F
A038:06 10 AF 77 CD 00 A0 10:C8
A040:F9 C9 7C D6 08 67 CB 74:F6
A048:C0 D5 11 B0 3F 19 D1 C9:15
A050:55 E1 DD 6E 00 DD 66 01:56
A058:22 50 A0 C9 2A 50 A0 CD:38
A060:42 A0 CD 42 A0 CD 42 A0:AD
A068:CD 42 A0 22 50 A0 2A 50:B7
A070:A0 DD 5E 00 DD 56 01 CD:48
A078:19 A0 C9 2A 50 A0 CD 00:5F
A080:A0 CD 00 A0 CD 00 A0 CD:A3
A088:00 A0 C3 6B A0 00 00 00:7D
A090:11 00 A4 DD 6E 00 DD 66:02
A098:01 06 10 C5 E5 06 04 7E:32
A0A0:12 13 23 10 FA E1 CD 00:E4
A0A8:A0 C1 10 EF C9 00 00 00:43
```

Alignez-vous ...

MANO NÉGRA

*Savez-vous réussir un Yol?
En d'autres termes, êtes-vous capable d'aligner dans un sens quelconque, 3 Zors de couleur identique?*



Ce jeu de pure stratégie, réclame un sang-froid hors pair. L'action consiste à disposer judicieusement, afin de constituer des alignements, les pions multicolores qui chutent de rangées verticales. Ne vous y fiez pas, on est très vite débordé!..

Sauvegarde

Sauvez sous un nom de votre choix, le programme Basic principal. Entrez ensuite par Amsaisie V.2 en vous reportant à son mode d'emploi, les deux listings de codes hexadécimaux.

Nom	Adr. déb.	Long
MANOSPRI	&8000	&D68
MANOROUT	&9D20	&35C

La longueur est ici précisée à l'attention de ceux qui envisagent de morceler leur travail en plusieurs fichiers qui devront ultérieurement être réunis en deux fichiers définitifs.

Claude Le Moullec

```
10 REM  :::::::::::::::::::::::           [1736]
20 REM  :                     :           [419]
30 REM  :       MANO NEGRA    :           [619]
40 REM  :                     :           [419]
50 REM  :::::::::::::::::::::::           [1736]
60 REM  :                     :           [419]
70 REM  :    LMC  SOFTWARE    :           [538]
80 REM  :  Claude LE MOULLEC  :           [2031]
90 REM  :                     :           [419]
100 REM :::::::::::::::::::::::           [1736]
110 REM :                     :           [419]
120 REM :     REDEFINITION    :           [1622]
130 REM :                     :           [419]
140 REM :::::::::::::::::::::::           [1736]
150 SYMBOL AFTER 169                      [1435]
160 SYMBOL 170,170,170,170,170,17         [2823]
0,170,170,170
170 SYMBOL 171,0,84,84,84,84,84,8         [1597]
4,0
180 SYMBOL 201,0,0,0,0,1,1,0,1            [1667]
190 SYMBOL 202,14,31,123,245,226,         [2905]
255,127,103
200 SYMBOL 203,128,224,192,224,25         [3220]
2,255,255,224
210 SYMBOL 204,0,0,0,58,239,255,2         [2001]
54,188
220 SYMBOL 205,3,3,1,0,1,3,15,62          [1814]
230 SYMBOL 206,108,113,191,220,17         [2683]
9,119,127,248
240 SYMBOL 207,15,252,128,0,192,2         [2530]
24,253,195
250 SYMBOL 208,240,0,0,0,0,0,0,12         [1947]
8
260 SYMBOL 209,124,249,0,0,0,0,0,         [1555]
0
270 SYMBOL 210,231,223,0,0,0,0,0,         [2016]
0
280 SYMBOL 211,63,255,0,0,0,0,0,0         [2345]
290 SYMBOL 212,192,192,0,0,0,0,0,         [1836]
0
300 SYMBOL 213,254,188,94,47,255,         [2394]
255,126,192
310 SYMBOL 214,254,252,30,15,255,         [2281]
193,0,0
320 MEMORY &7FFF:LOAD "|MANOSPRI"         [1098]
,&8000
330 LOAD "|MANOROUT",&9D20                [1427]
340 REM :::::::::::::::::::::::           [1736]
350 REM :                     :           [419]
360 REM :  VARIABLES DE BASE  :           [2081]
370 REM :                     :           [419]
380 REM :::::::::::::::::::::::           [1736]
390 DEFINT a-z:MODE 0:BORDER 0            [1453]
400 RESTORE 410:FOR h=0 TO 15:REA         [1813]
D a:INK h,a:NEXT
410 DATA 0,1,15,5,3,22,14,7,16,9,         [1985]
13,24,18,23,26,6
420 DEF FN pt(x,y)=&C000+(y-1)*80         [2454]
+(x-1)*2
430 DEF FN po(x,y)=&E003+(y-1)*80         [1359]
+(x-1)*7
440 DIM sp(15):FOR h=0 TO 10:sp(h         [3664]
```

```
+1)=&8BEE+(h*42):NEXT
45Ø DIM je(9,26):DIM cx(2ØØØ):DIM [2824]
 cy(2ØØØ):DIM cp(2ØØØ)
46Ø FOR H=1 TO 5:JE(H,19)=1Ø:NEXT [1313]
47Ø FOR h=1 TO 9:je(h,25)=1Ø:je(h [336Ø]
,26)=1Ø:je(h,19)=1Ø:je(h,18)=1Ø:N
EXT
48Ø FOR h=18 TO 26:je(1,h)=1Ø:je( [377Ø]
2,h)=1Ø:je(9,h)=1Ø:je(8,h)=1Ø:NEX
T
49Ø ali3$=CHR$(2Ø9)+CHR$(21Ø)+CHR [2257]
$(211)+CHR$(212)
5ØØ ali2$=CHR$(2Ø5)+CHR$(2Ø6)+CHR [2118]
$(2Ø7)+CHR$(2Ø8)
51Ø ali1$=CHR$(2Ø1)+CHR$(2Ø2)+CHR [3744]
$(2Ø3)+CHR$(2Ø4)
52Ø tr$=CHR$(22)+CHR$(1):nr$=CHR$ [1849]
(22)+CHR$(Ø)
53Ø ENT 2,2ØØ,2Ø,5:BUT=Ø:VIE=3    [2535]
54Ø ON BREAK GOSUB 285Ø           [755]
55Ø REM ::::::::::::::::::::::::  [1736]
56Ø REM :                      :  [419]
57Ø REM :     DESSIN DECOR     :  [1958]
58Ø REM :                      :  [419]
59Ø REM ::::::::::::::::::::::::  [1736]
6ØØ POKE &AØ1A,165:POKE &AØ1E,16: [2681]
CALL &AØØD,FN pt(2,2)-1,&8ØØØ
61Ø POKE &AØ1A,69:POKE &AØ1E,6:CA [2723]
LL &AØØD,FN pt(27,,5),&8A5Ø
62Ø FOR h=128 TO 14Ø STEP 4:PLOT  [2757]
h,Ø,Ø:DRAW h,4ØØ:NEXT
63Ø PLOT 128,1,1Ø:DRAW 128,398:DR [5171]
AW 414,398:DRAW 414,1:DRAW 128,1:
PLOT 128,3:DRAW 414,3:PLOT 128,39
6:DRAW 414,396
64Ø PLOT 132,5,6:DRAW 132,394:DRA [3547]
W 41Ø,394:DRAW 41Ø,5:DRAW 132,5:P
LOT 132,7:DRAW 41Ø,7:PLOT 132,392
:DRAW 41Ø,392
65Ø PLOT -1Ø,-1Ø,11:TAG:MOVE 498, [5Ø78]
368:PRINT ali1$;:MOVE 498,352:PRI
NT ali2$;:MOVE 498,336:PRINT ali3
$;:TAGOFF
66Ø DATA 16,1,5,16,5,5,16,8,5,16, [2885]
1Ø,5,16,13,5,16,15,5,18,18
67Ø DATA 3,18,2Ø,3,16,22,3,16,24, [2513]
3,16,1,5,2Ø,1,5,16,8,3,2Ø
68Ø DATA 8,3,16,13,3,2Ø,13,3,18,1 [2824]
8,3,2Ø,18,3,16,22,3,18,22,3
69Ø e1=1Ø:e2=6:RESTORE 66Ø:FOR i= [383Ø]
1 TO 2Ø:GOSUB 7ØØ:NEXT:GOTO 78Ø
7ØØ READ X,Y,L:x1=14+(X-1)*32:y1= [2428]
388-(Y-1)*16
71Ø IF i>1Ø THEN 75Ø              [1Ø58]
72Ø FOR J=Ø TO 2 STEP 2:PLOT x1,y [2455]
1+j,e1:DRAW x1+(l-1)*32,y1+j:NEXT
73Ø FOR J=4 TO 6 STEP 2:PLOT x1,y [4175]
1+j,e2:DRAW x1+(l-1)*32,y1+j:NEXT
74Ø RETURN                        [555]
75Ø PLOT x1+4,y1+2,e1:DRAW x1+4,y [2465]
1+2-(l-1)*16
76Ø PLOT x1,y1+2,e2:DRAW x1,y1+2- [1Ø96]
(l-1)*16
77Ø RETURN                        [555]
78Ø LOCATE 1,1:PRINT tr$:RESTORE  [4715]
79Ø:FOR h=1 TO 2Ø:GOSUB 81Ø:NEXT:
GOTO 83Ø
79Ø DATA 16,1,16,5,2Ø,1,2Ø,5,16,8 [3Ø14]
,2Ø,8,16,1Ø,2Ø,1Ø,16,13,2Ø,13
8ØØ DATA 16,15,2Ø,15,18,18,2Ø,18, [1874]
18,2Ø,2Ø,2Ø,16,22,18,22,16,24,18,
24
81Ø READ x,y:PEN 9:LOCATE x,y:PRI [3811]
NT CHR$(17Ø):PEN 12:LOCATE x,y
82Ø PRINT CHR$(171):RETURN        [1611]
83Ø RESTORE 84Ø:FOR h=1 TO 9:READ [44Ø7]
 ZL,YL,mo,a$:GOSUB 85Ø:NEXT:GOTO
94Ø
84Ø DATA 32,9,Ø,"ØØØØØØ",32,11,1, [7666]
"SCORE",32,14,Ø,"ØØØØØØ",36,19,Ø,
"ØØ",32,23,Ø,"ØØ",36,23,1,"YOL",3
Ø,19,Ø,"VIE",3Ø,6,1,"ALI GATOR",3
2,16,Ø,"RECORD"
85Ø A$=UPPER$(A$):FOR T=1 TO LEN( [27Ø6]
A$):SP=(ASC(MID$(A$,T,1)))-48
86Ø IF SP<Ø OR SP>43 THEN SP=43   [1734]
87Ø CALL &9D2Ø,FN PT(ZL,YL)+(T*2) [3746]
+MO,&9D4Ø+(SP*16):NEXT
88Ø RETURN                        [555]
89Ø REM ::::::::::::::::::::::::  [1736]
9ØØ REM :                      :  [419]
91Ø REM :   DEBUT DE PARTIE    :  [1477]
92Ø REM :                      :  [419]
93Ø REM ::::::::::::::::::::::::  [1736]
94Ø GOSUB 287Ø                    [9Ø1]
95Ø BUT=BUT+1                     [8Ø1]
96Ø TA=(BUT*7):TA1=TA:GOSUB 255Ø: [2127]
GOSUB 257Ø
97Ø VIT1=2ØØ-TA:IF VIT1<1ØØ THEN  [1775]
VIT1=1ØØ
98Ø VIT2=5Ø-(BUT*3):IF VIT2<27 TH [2424]
EN VIT2=27
99Ø GOSUB 283Ø:A$="EPREUVE NR"+ST [4188]
R$(BUT):ZL=11:YL=13:GOSUB 85Ø
1ØØØ RESTORE 272Ø:GOSUB 269Ø:FOR  [4133]
T=1 TO 4ØØØ:NEXT:GOSUB 283Ø:GOSUB
 279Ø
1Ø1Ø FIN=Ø:PER=Ø:PEL=Ø:POKE &AØ1A [2Ø75]
,7:POKE &AØ1E,6
1Ø2Ø CALL &AØØD,FN po(5,19),sp(9) [1614]
:xj=5:yj=16
1Ø3Ø c=INT(RND*8)+1:x=INT(RND*5)+ [4824]
3:CALL &AØØD,FN po(x,1),sp(c)
1Ø4Ø cx(1)=x:cy(1)=1:cp(1)=c:cc=1 [2462]
:dp=1
1Ø5Ø REM :::::::::::::::::::::::: [1736]
1Ø6Ø REM :                      : [419]
1Ø7Ø REM : ROUTINE PRINCIPALE   : [2225]
1Ø8Ø REM :                      : [419]
1Ø9Ø REM :::::::::::::::::::::::: [1736]
11ØØ EVERY VIT1,3 GOSUB 138Ø      [1529]
111Ø EVERY VIT2,2 GOSUB 141Ø      [18ØØ]
112Ø IF PEL=4 THEN 135Ø           [582]
113Ø IF fin=1 THEN DI:GOTO 227Ø   [689]
114Ø IF INKEY(8)*INKEY(74)=Ø AND  [2582]
xj>3 THEN 119Ø
115Ø IF INKEY(1)*INKEY(75)=Ø AND  [1457]
xj<7 THEN 122Ø
116Ø IF INKEY(9)*INKEY(76)=Ø THEN [1257]
 125Ø
117Ø GOTO 112Ø                    [347]
118Ø REM :::    A GAUCHE    :::   [1Ø52]
```

```
1190 DI:CALL &A04B,FN po(xj-1,yj)  [2724]
,FN po(xj,yj):xj=xj-1
1200 FOR t=1 TO 30:NEXT:SOUND 1,1  [1893]
00,2,5:EI:GOTO 1120
1210 REM :::    A DROITE    :::    [1375]
1220 DI:CALL &A04B,FN po(xj+1,yj)  [4305]
,FN po(xj,yj):xj=xj+1
1230 FOR t=1 TO 30:NEXT:SOUND 1,1  [1893]
00;2,5:EI:GOTO 1120
1240 REM :::    LACHER    :::      [317]
1250 DI:IF pel=0 THEN EI:GOTO 112  [2611]
0
1260 A=0:FOR H=1 TO PEL:IF JE(XJ,  [2488]
19+H)<>0 THEN A=1
1270 NEXT:IF A=1 THEN SOUND 1,150  [2659]
,15,7:EI:GOTO 1120
1280 CALL &A00D,FN po(xj,19),sp(9  [1635]
)
1290 FOR h=1 TO pel:p=pl(h):CALL   [2733]
&A00D,FN po(xj,19-h),SP(P)
1300 FOR g=20 TO 24:IF je(xj,g)=0  [1946]
 THEN a=g
1310 NEXT:CALL &A00D,FN po(xj,a),  [2305]
sp(p):je(xJ,a)=p
1320 GOSUB 1620:NEXT h:CALL &A00D  [3052]
,FN po(xj,19),sp(9)
1330 PEL=0:EI:GOTO 1120            [845]
1340 REM 4 ZORS SUR LA RAQUETTE    [1258]
1350 A=0:FOR H=1 TO 4:IF JE(XJ,19  [2639]
+H)<>0 THEN A=1
1360 NEXT:IF A=1 THEN fin=1:PER=1  [4316]
:GOTO 1280 ELSE 1280
1370 REM : APPARITION D'UN ZOR :   [1858]
1380 DI:c=INT(RND*8)+1:x=INT(RND*  [2932]
5)+3:CALL &A00D,FN po(x,1),sp(c)
1390 cc=cc+1:cx(cc)=x:cy(cc)=1:cp  [1885]
(cc)=c:EI:RETURN
1400 REM ::  CHUTE DES ZORS   ::   [2353]
1410 DI:FOR h=dp TO cc             [874]
1420 x1=cx(h):y1=cy(h):p=cp(h)     [1785]
1430 CALL &A032,FN po(x1,y1):y1=y  [2269]
1+1:IF y1=16 THEN 1460
1440 CALL &A00D,FN po(x1,y1),sp(p  [2326]
):cy(h)=y1
1450 NEXT:EI:RETURN                [861]
1460 IF x1=xj THEN 1520            [1206]
1470 IF je(x1,20)<>0 THEN CALL &A  [4107]
00D,FN po(x1,19),sp(p):fin=1:PER=
1:GOTO 1450
1480 dp=dp+1:FOR g=20 TO 24:IF je  [4027]
(x1,g)=0 THEN a=g
1490 NEXT:CALL &A00D,FN po(x1,a),  [3274]
sp(p):je(x1,a)=p
1500 GOSUB 1610:GOTO 1450          [1308]
1510 REM  CHUTE SUR LA RAQUETTE    [1917]
1520 dp=dp+1:pel=pel+1             [2473]
1530 CALL &A00D,FN po(x1,19-pel),  [3002]
sp(p):pl(pel)=p:GOTO 1450
1540 REM :::::::::::::::::::::::   [1736]
1550 REM :                     :   [419]
1560 REM :   ALGORITHME POUR   :   [1312]
1570 REM : CALCUL D'ALIGNEMENT :   [1039]
1580 REM :    (quel boulot !)  :   [1291]
1590 REM :                     :   [419]
1600 REM :::::::::::::::::::::::   [1736]
1610 XF=X1:YF=A:GOTO 1630          [1419]
1620 XF=XJ:YF=A                    [937]
1630 IF JE(XF,YF+1)=P AND JE(XF,Y  [2102]
F+2)=P THEN 1820
1640 IF JE(XF+1,YF)=P AND JE(XF+2  [2007]
,YF)=P THEN 1740
1650 IF JE(XF-1,YF)=P AND JE(XF-2  [3039]
,YF)=P THEN 1780
1660 IF JE(XF-1,YF)=P AND JE(XF+1  [2748]
,YF)=P THEN XF=XF-1:GOTO 1740
1670 IF JE(XF-1,YF-1)=P AND JE(XF  [2284]
-2,YF-2)=P THEN 1850
1680 IF JE(XF+1,YF+1)=P AND JE(XF  [1491]
+2,YF+2)=P THEN 1890
1690 IF JE(XF+1,YF-1)=P AND JE(XF  [2365]
+2,YF-2)=P THEN 1930
1700 IF JE(XF-1,YF+1)=P AND JE(XF  [1770]
-2,YF+2)=P THEN 1970
1710 IF JE(XF+1,YF-1)=P AND JE(XF  [1820]
-1,YF+1)=P THEN 2010
1720 IF JE(XF-1,YF-1)=P AND JE(XF  [2983]
+1,YF+1)=P THEN 2050
1730 RETURN                        [555]
1740 JE(XF,YF)=0:COX=XF:GOSUB 214  [1626]
0
1750 JE(XF+1,YF)=0:COX=XF+1:GOSUB  [1592]
 2140
1760 JE(XF+2,YF)=0:COX=XF+2:GOSUB  [1019]
 2140
1770 TA=TA-1:GOSUB 2570:IF TA=0 T  [2763]
HEN FIN=1:RETURN ELSE RETURN
1780 JE(XF,YF)=0:COX=XF:GOSUB 214  [1626]
0
1790 JE(XF-1,YF)=0:COX=XF-1:GOSUB  [2525]
 2140
1800 JE(XF-2,YF)=0:COX=XF-2:GOSUB  [2612]
 2140
1810 TA=TA-1:GOSUB 2570:IF TA=0 T  [2763]
HEN FIN=1:RETURN ELSE RETURN
1820 JE(XF,YF)=0:JE(XF,YF+1)=0:JE  [3254]
(XF,YF+2)=0
1830 COX=XF:GOSUB 2140             [1167]
1840 TA=TA-1:GOSUB 2570:IF TA=0 T  [2763]
HEN FIN=1:RETURN ELSE RETURN
1850 JE(XF,YF)=0:COX=XF:GOSUB 214  [1626]
0
1860 JE(XF-1,YF-1)=0:COX=XF-1:GOS  [1845]
UB 2140
1870 JE(XF-2,YF-2)=0:COX=XF-2:GOS  [2975]
UB 2140
1880 TA=TA-1:GOSUB 2570:IF TA=0 T  [2763]
HEN FIN=1:RETURN ELSE RETURN
1890 JE(XF,YF)=0:COX=XF:GOSUB 214  [1626]
0
1900 JE(XF+1,YF+1)=0:COX=XF+1:GOS  [1730]
UB 2140
1910 JE(XF+2,YF+2)=0:COX=XF+2:GOS  [1552]
UB 2140
1920 TA=TA-1:GOSUB 2570:IF TA=0 T  [2763]
HEN FIN=1:RETURN ELSE RETURN
1930 JE(XF,YF)=0:COX=XF:GOSUB 214  [1626]
0
1940 JE(XF+1,YF-1)=0:COX=XF+1:GOS  [2692]
UB 2140
1950 JE(XF+2,YF-2)=0:COX=XF+2:GOS  [2434]
UB 2140
1960 TA=TA-1:GOSUB 2570:IF TA=0 T  [2763]
```

```
HEN FIN=1:RETURN ELSE RETURN
1970 JE(XF,YF)=0:COX=XF:GOSUB 214 [1626]
0
1980 JE(XF-1,YF+1)=0:COX=XF-1:GOS [3337]
UB 2140
1990 JE(XF-2,YF+2)=0:COX=XF-2:GOS [2641]
UB 2140
2000 TA=TA-1:GOSUB 2570:IF TA=0 T [2763]
HEN FIN=1:RETURN ELSE RETURN
2010 JE(XF,YF)=0:COX=XF:GOSUB 214 [1626]
0
2020 JE(XF+1,YF-1)=0:COX=XF+1:GOS [2692]
UB 2140
2030 JE(XF-1,YF+1)=0:COX=XF-1:GOS [3337]
UB 2140
2040 TA=TA-1:GOSUB 2570:IF TA=0 T [2763]
HEN FIN=1:RETURN ELSE RETURN
2050 JE(XF,YF)=0:COX=XF:GOSUB 214 [1626]
0
2060 JE(XF-1,YF-1)=0:COX=XF-1:GOS [1845]
UB 2140
2070 JE(XF+1,YF+1)=0:COX=XF+1:GOS [1730]
UB 2140
2080 TA=TA-1:GOSUB 2570:IF TA=0 T [2763]
HEN FIN=1:RETURN ELSE RETURN
2090 REM :::::::::::::::::::::::: [1736]
2100 REM :                      : [419]
2110 REM : MISE A JOUR COLONNE  : [1137]
2120 REM :                      : [419]
2130 REM :::::::::::::::::::::::: [1736]
2140 POKE &A06F,40:CALL &A06E,FN  [2863]
po(COX,20):POKE &A06F,32
2150 G1=0:FOR I=24 TO 20 STEP -1  [1692]
2160 IF JE(COX,I)=0 THEN 2170 ELS [3330]
E G1=G1+1:G(G1)=JE(COX,I)
2170 NEXT                         [350]
2180 IF G1=0 THEN RETURN          [1135]
2190 FOR H5=20 TO 24:JE(COX,H5)=0 [2882]
:NEXT
2200 FOR I=1 TO G1:CALL &A00D,FN  [2705]
PO(COX,25-I),SP(G(I))
2210 JE(COX,25-I)=G(I):NEXT:RETUR [1235]
N
2220 REM :::::::::::::::::::::::: [1736]
2230 REM :                      : [419]
2240 REM :     FIN DE PARTIE    : [858]
2250 REM :                      : [419]
2260 REM :::::::::::::::::::::::: [1736]
2270 FOR H=0 TO 3:MU=REMAIN(H):NE [2096]
XT
2280 RESTORE 2670:GOSUB 2690      [1762]
2290 ERASE JE,CX,CY,CP:DIM je(9,2 [2724]
6):DIM cx(2000):DIM cy(2000):DIM
cp(2000)
2300 IF PER=1 THEN 2350           [911]
2310 FOR H=1 TO TA1:SC=SC+5:ENV 1 [3429]
0,15,-1,1:SOUND 1,0,15,15,10,,15
2320 GOSUB 2430:NEXT:IF SC>REC TH [3948]
EN REC=SC:GOSUB 2490
2330 EI:GOTO 950                  [321]
2340 REM :::   PERDU    :::       [1027]
2350 VIE=VIE-1:BUT=BUT-1:IF SC>RE [3288]
C THEN REC=SC:GOSUB 2490
2360 SC=0:GOSUB 2430:IF VIE=0 THE [1800]
N BUT=0:VIE=3
2370 EI:GOTO 950                  [321]
2380 REM ::::::::::::::::::::::: [1661]
2390 REM :                     : [419]
2400 REM :   GESTION SCORES    : [1349]
2410 REM :                     : [419]
2420 REM ::::::::::::::::::::::: [1661]
2430 IF sc=0 THEN a$="000000":ZL= [2483]
32:YL=9:GOSUB 850:RETURN
2440 a$=STR$(sc):A$=RIGHT$(A$,LEN [2868]
(A$)-1)
2450 IF sc<10 THEN ZL=35:YL=9:GOS [3048]
UB 850:RETURN
2460 IF sc<100 THEN ZL=34:YL=9:GO [2041]
SUB 850:RETURN
2470 IF sc<1000 THEN ZL=33:YL=9:G [2172]
OSUB 850:RETURN
2480 ZL=32:YL=9:GOSUB 850:RETURN  [2194]
2490 IF REC=0 THEN a$="000000":ZL [2264]
=32:YL=14:GOSUB 850:RETURN
2500 a$=STR$(REC):A$=RIGHT$(A$,LE [2806]
N(A$)-1)
2510 IF REC<10 THEN ZL=35:YL=14:G [2332]
OSUB 850:RETURN
2520 IF REC<100 THEN ZL=34:YL=14: [3185]
GOSUB 850:RETURN
2530 IF REC<1000 THEN ZL=33:YL=14 [1937]
:GOSUB 850:RETURN
2540 ZL=32:YL=14:GOSUB 850:RETURN [1660]
2550 a$=STR$(vie):A$=RIGHT$(A$,LE [1991]
N(A$)-1)
2560 ZL=37:YL=19:GOSUB 850:RETURN [2079]
2570 a$=STR$(ta):A$=RIGHT$(A$,LEN [1517]
(A$)-1)
2580 SOUND 1,150,50,7,0,2:IF ta<1 [2185]
0 THEN A$=" "+A$
2590 IF TA<100 THEN ZL=32:YL=23:G [3340]
OSUB 850:RETURN
2600 ZL=31:YL=23:GOSUB 850:RETURN [2629]
2610 REM ::::::::::::::::::::::: [1661]
2620 REM :                     : [419]
2630 REM :  SOUS PROG DIVERS   : [1142]
2640 REM :                     : [419]
2650 REM ::::::::::::::::::::::: [1661]
2660 REM :::    MUSIQUES    :::  [1065]
2670 DATA 1,213,2,1,0,1,1,213,1,1 [5592]
,0,1,1,213,1,1,0,1,17,169,1,10,21
3,1,1,0,1,1,213,1,1,0,1,17,169,1,
10,213,1,1,0,1,49,142,1,42,169,1,
28,213,1
2680 DATA 1,0,1,49,106,2,42,142,2 [6794]
,28,169,2,1,0,1,49,106,1,42,142,1
,28,169,1,1,0,1,49,106,1,42,142,1
,28,169,1,1,0,1,49,106,4,42,142,4
,28,169,4,1,0,1,0,0,0
2690 ca=1:WHILE ca<>0             [1244]
2700 READ ca,p,z:SOUND ca,p,z*10, [2058]
15
2710 WEND:RETURN                  [935]
2720 DATA 49,159,2,42,190,2,28,23 [1045]
9,2
2730 DATA 17,190,1,10,239,1,49,17 [7440]
9,2,42,213,2,28,239,2,49,159,4,42
,190,4,28,239,4,49,159,2,42,190,2
,28,239,2,49,142,2,42,179,2,28,23
9,2,49,159,2,42,190,2,28,239,2,49
,142,2,42,179,2
2740 DATA 28,239,2,49,134,2,42,15 [8999]
```

```
9,2,28,2Ø1,2,17,159,1,1Ø,2Ø1,1,49
,15Ø,1,42,179,1,28,2Ø1,1,49,134,4
,42,159,4,28,2Ø1,4,49,134,2,42,15
9,2,28,2Ø1,2,49,119,2,42,15Ø,2,28
,2Ø1,2,49,134,2,42,159,2
275Ø DATA 28,2Ø1,2,49,119,2,42,15 [6855]
Ø,2,28,2Ø1,2,49,113,4,42,134,4,28
,169,4,49,113,1,42,134,1,28,169,1
,49,134,1,42,169,1,28,239,1,49,11
3,2,42,134,2,28,169,2,49,1Ø6,4,42
,134,4,28,179,4
276Ø DATA 49,1Ø6,1,42,134,1,28,17 [7892]
9,1,49,134,1,42,179,1,28,213,1,49
,1Ø6,2,42,134,2,28,179,2,49,1Ø6,4
,42,127,4,28,159,4,49,1Ø6,1,42,12
7,1,28,159,1,49,127,1,42,159,1,28
,213,1
277Ø DATA 49,1Ø6,2,42,127,2,28,15 [7693]
9,2,49,1ØØ,4,42,127,4,28,169,4,49
,1ØØ,1,42,127,1,28,169,1,49,127,1
,42,169,1,28,2Ø1,1,49,1ØØ,2,42,12
7,2,28,169,2,49,95,1Ø,42,127,1Ø,2
8,15Ø,1Ø,Ø,Ø,Ø
278Ø REM  AFFICHER AIRE DE JEU    [1144]
279Ø FOR h=184 TO 4ØØ STEP 56:PLO [3784]
T h,392,9:DRAW h,9:PLOT h+4,392,1
2:DRAW h+4,9:NEXT
28ØØ FOR g=376 TO 15Ø STEP -16:FO [345Ø]
R h=156 TO 4ØØ STEP 56:PLOT h,g,1
3:DRAW h+8,g:NEXT h,g
281Ø RETURN                            [555]
282Ø REM   EFFACE AIRE DE JEU      [1921]
283Ø POKE &AØ39,192:POKE &AØ3D,34  [1552]
284Ø CALL &AØ32,&EØ11:POKE &AØ39,  [3Ø7Ø]
7:POKE &AØ3D,6:RETURN
285Ø FOR MU=Ø TO 3:L=REMAIN(MU):N  [4928]
EXT:MODE 2:INK 1,24:PEN 1:END
286Ø REM ::   EXPLICATIONS   ::     [137Ø]
287Ø GOSUB 283Ø:RESTORE 272Ø:GOSU  [1691]
B 269Ø
288Ø RESTORE 291Ø:FOR h=1 TO 12:R  [2666]
EAD a$
289Ø zl=9:yl=h*2:GOSUB 85Ø          [2127]
29ØØ NEXT:CALL &BB18:RETURN         [1238]
291Ø DATA "   MANO NEGRA   "       [1243]
292Ø DATA ""                         [2Ø8]
293Ø DATA " UN YOL CE N>EST"        [1288]
294Ø DATA "QUE 3 ZORS DE LA"        [1Ø24]
295Ø DATA "MEME COULEUR QUE"        [12Ø3]
296Ø DATA "TU ALIGNES  DANS"        [319]
297Ø DATA "LE  SENS  QUE TU"        [1175]
298Ø DATA "VEUX<"                    [426]
299Ø DATA " COMBIEN DE YOLS"         [275]
3ØØØ DATA "FERAS-TU ?"               [916]
3Ø1Ø DATA ""                         [2Ø8]
3Ø2Ø DATA "  BONNE CHANCE"          [1147]
```

```
8000:00 00 00 00 00 00 00 00:80
8008:00 00 00 00 00 00 00 00:88
8010:00 00 00 00 00 00 00 00:90
8018:00 00 00 00 00 00 00 00:98
8020:00 00 00 00 00 00 00 00:A0
8028:00 00 00 10 30 20 00 00:98
8030:00 00 00 03 00 00 00 00:B0
8038:00 00 00 30 03 30 00 00:A7
8040:30 03 03 03 00 00 00 00:C0
8048:00 00 10 03 03 12 20 00:5F
8050:00 00 00 00 00 00 00 00:D0
8058:00 00 30 03 03 12 20 00:CF
8060:00 00 00 00 00 00 00 00:E0
8068:00 00 21 03 03 03 20 00:58
8070:00 00 00 00 00 00 00 00:F0
8078:00 00 21 03 03 03 30 00:D8
8080:00 00 00 00 00 00 00 00:00
8088:00 10 03 03 03 03 30 00:AE
8090:00 00 00 00 00 00 00 00:10
8098:00 10 03 03 03 03 12 00:EC
80A0:00 00 00 00 00 00 00 00:20
80A8:00 01 03 03 03 03 12 00:DE
80B0:00 00 00 00 00 00 00 00:30
80B8:00 01 03 03 03 03 12 20:EE
80C0:00 00 00 00 00 00 00 00:40
80C8:00 01 03 03 03 03 12 20:FE
80D0:00 00 00 00 00 00 00 00:50
80D8:00 01 03 03 03 03 12 20:0E
80E0:00 00 00 00 00 00 00 00:60
80E8:00 01 03 03 03 03 12 30:9E
80F0:00 00 00 00 00 00 00 00:70
80F8:00 03 03 03 03 03 12 30:B2
8100:00 00 00 00 00 00 00 00:81
8108:00 03 03 03 03 03 12 30:C3
8110:00 00 00 00 00 00 00 00:91
8118:00 03 03 03 03 03 12 30:D3
8120:00 00 00 00 00 00 00 00:A1
8128:00 03 03 03 03 03 12 30:E3
8130:00 00 00 00 00 00 00 00:B1
8138:00 21 03 03 03 03 12 30:2F
8140:00 00 00 00 00 00 00 00:C1
8148:00 21 03 03 03 03 12 30:3F
8150:00 00 00 00 00 00 00 00:D1
8158:00 12 03 03 03 03 30 30:03
8160:00 00 00 00 00 00 00 00:E1
8168:00 03 03 03 03 03 30 20:75
8170:00 00 00 00 00 00 00 00:F1
8178:00 03 03 03 03 03 30 20:85
8180:00 00 00 00 00 00 00 00:01
8188:00 03 03 03 03 03 30 20:95
8190:00 00 00 00 00 00 00 00:11
8198:00 03 03 03 03 03 30 20:A5
81A0:00 00 00 00 00 00 00 00:21
81A8:00 03 03 03 03 12 30 00:0F
81B0:00 00 00 00 00 00 00 00:31
81B8:00 21 03 03 03 12 30 00:5B
81C0:00 00 00 00 00 00 00 00:41
81C8:00 21 03 03 03 12 30 00:6B
81D0:00 00 00 00 00 00 00 00:51
81D8:10 21 03 03 03 03 30 00:31
81E0:00 00 00 00 00 00 00 00:61
81E8:10 21 21 21 21 03 20 00:39
81F0:00 00 00 00 00 10 30 20:21
81F8:10 21 12 12 03 03 02 00:78
8200:00 00 00 00 00 21 03 03:75
8208:12 21 21 21 21 03 12 00:FA
8210:00 00 00 00 10 03 03 03:21
8218:03 21 12 12 03 03 12 20:FC
8220:00 00 00 00 21 03 03 03:86
8228:03 03 03 21 21 03 12 20:75
8230:00 00 00 00 21 03 03 03:96
8238:03 03 12 12 03 03 12 20:E0
8240:00 00 00 10 21 03 03 07:06
8248:03 03 03 21 03 03 30 20:D1
8250:00 00 00 10 03 03 03 0B:A0
8258:03 03 03 12 03 03 30 30:25
8260:00 00 00 21 03 03 03 03:B4
8268:03 03 03 03 03 12 30 30:53
8270:00 00 00 21 03 03 03 03:C4
8278:03 03 03 03 03 30 30 30:17
8280:00 00 10 03 03 03 07 03:A8
8288:03 03 03 03 12 30 30 30:72
8290:00 00 10 12 03 03 0B 03:10
8298:03 03 03 03 30 30 30 30:18
82A0:00 00 21 12 03 07 03 03:33
82A8:03 03 03 03 12 30 30 30:92
82B0:00 00 21 03 03 03 03 03:EF
82B8:03 03 03 03 03 30 30 30:57
82C0:00 00 03 03 03 03 03 03:A5
82C8:03 03 03 03 03 30 30 30:67
82D0:00 10 03 21 03 07 03 03:65
82D8:03 03 03 03 03 12 30 30:C3
82E0:00 10 03 03 03 03 03 03:E5
82E8:03 03 03 03 03 03 30 30:79
82F0:00 10 03 12 03 03 03 03:31
82F8:03 03 03 03 03 03 30 30:89
8300:00 10 03 03 03 03 03 03:06
8308:03 03 03 03 03 03 30 30:9A
8310:00 30 03 03 12 03 03 03:A1
8318:03 03 03 03 30 03 30 30:8B
8320:00 30 03 12 21 21 03 03:EC
8328:03 03 03 12 0F 21 21 20:FD
8330:00 21 03 12 12 03 03 03:DF
8338:03 03 03 25 0F 1A 12 00:C6
8340:00 21 03 03 03 21 03 03:1C
8348:03 03 12 25 0F 0F 21 20:2A
8350:00 21 03 03 30 12 03 03:B3
8358:03 03 25 0F 0F 0F 21 02:2B
8360:00 21 03 03 21 21 03 03:D2
8368:03 03 25 0F 0F 0F 1A 02:0A
8370:10 21 03 03 12 12 03 03:4D
8378:03 12 0F 0F 0F 0F 1A 02:E6
8380:10 21 03 03 03 21 21 03:3E
8388:03 12 0F 0F 0F 0F 1A 03:0E
8390:10 03 03 03 12 12 12 03:9A
8398:03 25 0F 0F 0F 0F 0F 21:E7
83A0:10 03 03 03 12 21 21 03:6D
83A8:03 25 0F 0F 0F 0F 0F 21:F7
83B0:10 03 03 03 12 30 12 03:6E
83B8:03 25 0F 0F 0F 0F 0F 21:07
83C0:10 03 03 03 12 21 30 03:F6
83C8:03 25 0F 0F 0F 0F 0F 21:17
83D0:10 03 03 03 03 12 30 21:51
83D8:03 12 0F 0F 0F 0F 1A 03:5E
83E0:30 03 03 03 03 21 03 03:B0
83E8:03 12 0F 0F 0F 0F 1A 03:6E
83F0:30 03 03 03 12 03 03 03:57
83F8:03 03 25 0F 0F 0F 21 03:D3
8400:30 03 03 03 21 03 03 03:B3
8408:03 03 12 0F 0F 1A 03 03:1B
8410:30 03 03 03 21 03 03 03:C3
8418:03 03 03 25 0F 21 03 03:80
8420:30 03 03 12 21 03 03 07:2F
8428:03 03 03 25 0F 21 03 03:90
8430:30 03 03 12 03 03 03 0B:C9
8438:03 03 03 12 1A 03 03 02:CF
8440:30 21 03 21 03 03 03 03:11
8448:03 03 03 03 30 03 03 02:11
8450:30 30 03 21 03 03 03 03:3F
8458:03 03 03 03 03 03 03 02:40
8460:30 30 30 03 03 03 07 03:7A
8468:03 03 03 03 03 03 03 00:40
8470:30 30 30 12 03 03 0B 03:E2
8478:03 03 03 03 03 03 03 00:50
8480:30 30 21 12 03 07 03 03:A5
8488:03 03 03 03 03 03 20 00:2B
8490:10 30 21 03 03 03 03 03:41
8498:03 03 03 03 03 00 00 00:49
84A0:10 30 03 03 03 03 03 03:F7
84A8:03 03 03 03 03 00 00 00:59
84B0:10 30 03 21 03 07 03 03:97
84B8:03 03 03 03 03 02 00 00:75
84C0:10 30 03 03 03 03 03 03:17
84C8:03 03 03 03 03 03 00 00:8B
84D0:10 30 03 12 03 03 03 03:63
84D8:03 03 03 03 03 03 00 00:9B
84E0:10 30 03 03 03 03 03 03:37
84E8:03 03 03 03 03 03 02 00:B9
84F0:00 30 03 03 12 03 03 03:82
84F8:03 03 03 03 30 03 02 00:AA
8500:00 30 03 12 21 21 03 03:CE
8508:03 03 03 12 0F 21 03 00:0D
8510:00 21 03 12 12 03 03 03:C1
8518:03 03 03 25 0F 1A 03 00:3F
8520:00 21 03 03 03 21 03 03:FE
8528:03 03 12 25 0F 0F 21 00:0C
8530:00 21 03 03 30 12 03 03:95
8538:03 03 25 0F 0F 0F 21 02:0D
8540:00 21 03 03 21 21 03 03:B4
8548:03 03 25 0F 0F 0F 1A 02:EC
8550:10 21 03 03 12 12 03 03:2F
8558:03 12 0F 0F 0F 0F 1A 02:D8
8560:10 21 03 03 03 21 21 03:20
8568:03 12 0F 0F 0F 0F 1A 03:F0
8570:10 03 03 03 12 12 12 03:7C
8578:03 25 0F 0F 0F 0F 0F 21:C9
8580:10 03 03 03 12 21 21 03:4F
8588:03 25 0F 0F 0F 0F 0F 21:D9
8590:10 03 03 03 12 30 12 03:50
8598:03 25 0F 0F 0F 0F 0F 21:E9
85A0:10 03 03 03 12 21 30 03:D8
85A8:03 25 0F 0F 0F 0F 0F 21:F9
85B0:10 03 03 03 03 12 30 21:33
85B8:03 12 0F 0F 0F 0F 1A 03:40
85C0:30 03 03 03 03 03 03 30:46
85C8:03 12 0F 0F 0F 0F 1A 03:50
85D0:30 03 03 03 03 03 03 12:66
85D8:21 03 25 0F 0F 0F 21 03:D3
85E0:30 03 03 03 03 03 03 03:FE
85E8:30 03 12 0F 0F 1A 03 03:29
85F0:30 03 03 03 03 03 03 12:86
85F8:10 03 03 25 0F 21 03 03:6E
8600:30 03 03 03 03 03 03 12:97
8608:20 21 03 25 0F 21 03 03:CB
8610:30 03 03 03 03 03 03 12:A7
8618:20 10 03 12 1A 03 03 02:E8
8620:30 21 03 03 03 03 03 12:F3
8628:20 10 21 03 30 03 03 02:84
8630:30 03 03 03 03 03 03 03:4F
8638:20 00 21 03 03 03 03 02:93
8640:30 03 30 03 03 03 03 03:E6
8648:30 00 10 03 03 03 03 00:70
8650:30 03 07 1A 03 03 03 03:D7
8658:30 00 10 03 03 03 03 00:80
8660:30 03 12 30 30 03 03 03:41
8668:30 00 00 21 03 03 20 00:A3
8670:10 03 12 30 0B 30 30 30:29
8678:30 20 00 10 30 30 30 00:0E
8680:10 30 03 03 03 1A 30 30:06
8688:30 20 01 30 30 30 12 20:CF
8690:10 21 03 03 03 30 03 03:D9
8698:12 30 12 0F 0F 0F 21 2A:8E
86A0:10 25 07 25 30 03 03 03:58
86A8:03 30 25 0F 0F 0F 21 30:48
86B0:00 30 12 30 03 03 03 03:DA
86B8:03 21 25 0F 0F 0F 1A 30:09
86C0:00 21 03 03 03 03 03 03:EB
86C8:12 21 25 0F 0F 0F 1A 30:28
86D0:00 30 03 03 03 03 03 03:19
86D8:30 03 25 0F 0F 0F 1A 30:1A
86E0:00 30 21 03 03 03 03 12:FB
86E8:21 03 25 0F 0F 0F 1A 30:1B
86F0:00 10 30 03 03 03 03 30:E8
86F8:21 03 12 0F 0F 0F 21 30:23
8700:00 10 30 03 03 03 03 30:F9
8708:03 03 12 0F 0F 0F 21 30:16
8710:00 10 30 21 03 03 12 30:EA
8718:03 03 02 30 30 30 03 30:13
```

```
8720:00 00 30 30 03 03 12 30:16
8728:03 03 03 03 03 03 03 20:03
8730:00 00 30 30 03 03 12 30:26
8738:03 03 03 03 03 03 03 20:13
8740:00 00 10 30 03 03 12 21:5E
8748:03 03 03 03 03 03 03 00:23
8750:00 00 10 30 21 03 12 21:04
8758:03 03 03 03 03 03 12 00:9C
8760:00 00 00 30 21 03 12 21:E4
8768:03 03 03 03 03 03 12 00:AC
8770:00 00 00 30 30 03 12 21:3F
8778:03 03 03 03 03 03 20 00:1E
8780:00 00 00 10 30 03 12 21:CF
8788:03 03 03 03 03 12 20 00:88
8790:00 00 00 10 30 21 30 03:75
8798:03 03 03 12 30 20 00 00:29
87A0:00 00 00 00 30 21 30 03:45
87A8:03 03 03 30 30 00 00 00:F1
87B0:00 00 00 00 30 21 30 03:55
87B8:03 03 03 30 20 00 00 00:B1
87C0:00 00 00 00 30 30 30 03:BF
87C8:03 03 12 30 20 00 00 00:EE
87D0:00 00 00 00 10 30 30 03:2F
87D8:03 03 12 30 20 00 00 00:FE
87E0:00 00 00 00 10 30 30 03:3F
87E8:03 03 30 30 20 00 00 00:68
87F0:00 00 00 00 10 30 30 07:6F
87F8:03 03 30 30 00 00 00 00:D8
8800:00 00 00 00 00 30 21 03:A7
8808:03 03 00 00 00 00 00 00:99
8810:00 00 00 00 00 30 21 03:B7
8818:0B 03 30 30 20 00 00 00:A1
8820:00 00 00 00 00 30 21 03:C7
8828:03 12 30 30 30 00 00 00:17
8830:00 00 00 00 00 10 21 03:17
8838:07 03 12 30 30 00 00 00:B3
8840:00 00 00 00 00 10 21 03:27
8848:03 03 03 30 30 00 00 00:92
8850:00 00 00 00 00 00 21 03:D7
8858:03 03 03 30 30 20 00 00:62
8860:00 00 00 00 00 00 21 03:E7
8868:03 03 03 12 30 20 00 00:FA
8870:00 00 00 00 00 00 30 03:60
8878:03 03 03 12 30 10 20 00:8A
8880:00 00 00 00 00 00 10 03:90
8888:03 03 03 12 20 30 30 00:7A
8890:00 00 00 00 00 00 10 03:A0
8898:03 03 03 12 10 30 30 20:3A
88A0:00 00 00 00 00 00 10 03:B0
88A8:03 03 03 21 12 30 30 30:10
```

```
88B0:00 00 00 00 00 00 10 21:B0
88B8:03 03 03 03 12 30 30 30:A8
88C0:00 00 00 00 00 00 00 21:50
88C8:03 03 07 03 12 30 30 30:C4
88D0:00 00 00 00 00 00 00 21:60
88D8:03 03 03 03 03 03 30 30:6F
88E0:00 00 00 00 00 00 00 30:E8
88E8:03 03 0B 03 03 03 12 30:C5
88F0:00 00 00 00 00 00 00 10:F8
88F8:03 03 03 03 03 03 03 30:54
8900:00 00 00 00 00 00 00 10:09
8908:03 03 03 03 03 03 03 03:FD
8910:00 00 00 00 00 00 00 00:99
8918:21 03 0B 03 03 03 03 03:43
8920:00 00 00 00 00 00 00 00:A9
8928:21 03 03 03 03 03 03 03:3B
8930:00 00 00 00 00 00 00 00:B9
8938:21 03 03 03 03 03 03 03:4B
8940:00 00 00 00 00 00 00 00:C9
8948:10 03 03 03 03 03 03 03:4A
8950:00 00 00 00 00 00 00 00:D9
8958:10 21 0B 03 03 03 03 03:AE
8960:00 30 00 00 00 00 00 00:E9
8968:00 21 03 03 03 03 03 03:96
8970:00 00 00 00 00 00 00 00:F9
8978:00 01 03 03 03 03 03 03:66
8980:00 00 00 00 00 00 00 00:09
8988:00 10 03 03 03 03 03 03:94
8990:00 00 00 00 00 00 00 00:19
8998:00 00 21 03 03 03 03 03:DE
89A0:00 00 00 00 00 00 00 00:29
89A8:00 00 21 03 03 03 03 03:EE
89B0:00 00 00 00 00 00 00 00:39
89B8:00 00 10 03 03 03 03 03:CB
89C0:00 00 00 00 00 00 00 00:49
89C8:00 00 00 21 03 03 03 03:23
89D0:00 00 00 00 00 00 00 00:59
89D8:00 00 00 10 03 03 03 03:EF
89E0:00 00 00 00 00 00 00 00:69
89E8:00 00 00 10 21 03 03 03:95
89F0:00 00 00 00 00 00 00 00:79
89F8:00 00 00 00 30 03 03 03:B0
8A00:00 00 00 00 00 00 00 00:8A
8A08:00 00 00 00 10 21 03 03:D5
8A10:00 00 00 00 00 00 00 00:9A
8A18:00 00 00 00 00 30 03 03:EF
8A20:00 00 00 00 00 00 00 00:AA
8A28:00 00 00 00 00 10 21 03:11
8A30:00 00 00 00 00 00 00 00:BA
8A38:00 00 00 00 00 00 30 30:92
```

```
8A40:00 00 00 00 00 00 00 00:CA
8A48:00 00 00 00 00 00 10 30:C2
8A50:00 30 20 00 00 00 10 03:22
8A58:12 00 00 00 21 03 03 20:C0
8A60:00 00 21 03 03 12 00 00:D4
8A68:21 03 03 12 20 00 21 03:09
8A70:03 12 0A 00 21 03 03 12:9B
8A78:0A 00 21 03 03 30 0A 00:F0
8A80:21 03 03 25 0A 00 21 03:FF
8A88:03 25 0A 00 21 03 03 25:71
8A90:0A 00 21 03 03 25 0A 00:C6
8A98:21 03 03 25 0A 00 21 03:17
8AA0:03 25 0A 00 21 03 03 25:89
8AA8:0A 00 21 03 03 25 0A 00:DE
8AB0:21 03 03 30 0A 00 21 03:5B
8AB8:03 12 0A 00 21 03 03 12:E3
8AC0:0A 00 21 03 03 12 20 00:1E
8AC8:21 03 03 03 30 00 21 03:7D
8AD0:03 03 12 00 21 03 03 03:7D
8AD8:03 20 21 03 03 03 03 20:4A
8AE0:21 03 03 03 03 20 21 03:74
8AE8:03 03 03 20 30 03 03 03:33
8AF0:03 20 10 03 03 03 03 20:2F
8AF8:10 21 03 03 03 20 10 03:40
8B00:03 03 03 30 10 12 03 03:46
8B08:03 12 30 03 21 03 03 12:B2
8B10:21 03 03 03 03 12 21 03:51
8B18:03 03 03 12 21 07 03 03:F9
8B20:03 12 21 0B 03 03 03 12:27
8B28:21 03 03 03 03 03 25 03:2B
8B30:03 03 03 03 21 03 03 03:BD
8B38:03 03 21 03 03 03 03 03:89
8B40:21 0B 03 03 03 12 21 03:91
8B48:03 03 03 03 25 03 03 03:E9
8B50:03 0B 21 03 03 03 03 03:B1
8B58:21 03 03 03 03 0B 25 03:8B
8B60:03 03 03 03 21 03 03 03:ED
8B68:03 03 21 0B 03 03 03 0B:19
8B70:21 03 03 03 03 03 21 0B:97
8B78:03 03 03 0B 21 03 03 03:25
8B80:07 03 21 03 03 03 03 0B:15
8B88:21 03 03 03 07 03 30 0B:2C
8B90:03 03 03 12 30 07 03 03:BC
8B98:0B 12 30 03 0B 03 07 30:E8
8BA0:10 21 03 07 03 20 10 21:E9
8BA8:0B 0B 12 20 10 21 03 03:4D
8BB0:12 00 10 03 07 03 30 00:0E
8BB8:30 03 0B 0B 20 00 30 07:EE
8BC0:03 12 00 00 30 03 07 12:35
8BC8:00 00 30 03 0B 30 00 00:46
```

```
8BD0:30 03 03 20 00 00 30 21:72
8BD8:12 20 00 00 30 21 30 00:BB
8BE0:00 00 30 30 20 00 00 00:5B
8BE8:10 30 00 00 00 00 55 FF:2E
8BF0:FF FF FF AA EE EE EE EE:49
8BF8:EE FF EE EE EE EE EE FF:A5
8C00:EE CC CC CC CC DD FF DD:B1
8C08:DD DD DD DD FF DD DD DD:52
8C10:DD DD 55 FF FF FF FF AA:6C
8C18:04 0C 0C 0C 0C 08 0C 0C:34
8C20:0C 0C 0C 0C 0C 0C 0C 0C:5C
8C28:0C 0C 18 30 30 30 30 24:60
8C30:0C 0C 0C 0C 0C 0C 0C 0C:6C
8C38:0C 0C 0C 0C 04 0C 0C 0C:4C
8C40:0C 08 44 CC CC CC CC 88:7C
8C48:CC CC CC CC CC CC CC CC:84
8C50:CC CC CC CC D8 F0 F0 F0:BC
8C58:F0 E4 CC CC CC CC CC CC:E8
8C60:CC CC CC CC CC CC 44 CC:E4
8C68:CC CC CC 88 50 F0 F0 F0:1C
8C70:F0 A0 F0 F0 F0 F0 F0 F0:1C
8C78:F0 F0 F0 F0 F0 F0 F4 FC:40
8C80:FC FC FC F8 F0 F0 F0 F0:34
8C88:F0 F0 F0 F0 F0 F0 F0 F0:D4
8C90:50 F0 F0 F0 F0 A0 54 FC:78
8C98:FC FC FC A8 FC FC FC FC:44
8CA0:FC FC FC FC FC FC FC FC:9C
8CA8:E9 C3 C3 C3 C3 D6 FC FC:8F
8CB0:FC FC FC FC FC FC FC FC:AC
8CB8:FC FC 54 FC FC FC FC A8:1C
8CC0:45 CF CF CF CF 8A CF CF:40
8CC8:CF CF CF CF CF CF CF CF:70
8CD0:CF CF DF FF FF FF FF EF:C8
8CD8:CF CF CF CF CF CF CF CF:80
8CE0:CF CF CF CF 45 CF CF CF:D6
8CE8:CF 8A 41 C3 C3 C3 C3 82:EC
8CF0:C3 C3 C3 C3 C3 C3 C3 C3:E8
8CF8:C3 C3 C3 C3 97 3F 3F 3F:40
8D00:3F 6B C3 C3 C3 C3 C3 C3:C5
8D08:C3 C3 C3 C3 C3 C3 41 C3:73
8D10:C3 C3 C3 82 40 C0 C0 C0:37
8D18:C0 80 C0 C0 C0 C0 C0 C0:25
8D20:C0 0C 0C 0C 0C C0 84 C0:31
8D28:C0 C0 C0 48 C0 0C 0C 0C:11
8D30:0C C0 C0 C0 C0 C0 C0 C0:09
8D38:40 C0 C0 C0 C0 80 37 3F:FE
8D40:3F 3F 3F 3B 3B 3C 3C 3C:46
8D48:3C 37 3E 66 CC CC 99 3D:AC
8D50:6E BB FF FF 77 9D 3E 66:9D
8D58:CC CC 99 3D 3B 3C 3C 3C:1B
8D60:3C 37 37 3F 3F 3F 3F 3B:7E
```

```
9D20:DD 5E 00 DD 56 01 DD 6E:F9
9D28:02 DD 66 03 06 08 C5 E5:98
9D30:06 02 1A 77 23 13 10 FA:62
9D38:E1 CD 26 BC C1 10 EF C9:A8
9D40:15 3F 15 3F 45 45 45 45:AD
9D48:04 04 04 04 55 FF 00 00:B0
9D50:00 2A 15 2A 00 8A 00 8A:B4
9D58:00 08 00 08 55 FF 00 00:C8
9D60:15 3F 15 15 00 45 00 45:E9
9D68:00 08 04 00 55 FF 00 00:C4
9D70:15 3F 15 15 00 45 00 45:F9
9D78:00 0C 00 04 55 FF 00 00:E0
9D80:00 15 00 3F 00 CF 45 45:28
9D88:04 0C 00 04 00 55 00 00:4F
9D90:15 3F 15 00 45 00 00 CF:D0
9D98:00 04 04 04 55 FF 00 00:FC
9DA0:15 3F 15 15 45 00 45 CF:17
9DA8:04 04 04 04 55 FF 00 00:10
9DB0:15 3F 00 15 00 45 00 8A:22
9DB8:00 08 04 00 55 00 00 00:1A
9DC0:15 3F 15 15 45 45 45 CF:D5
9DC8:04 04 04 04 55 FF 00 00:30
9DD0:15 3F 15 15 45 45 45 CF:E5
9DD8:00 04 00 04 55 FF 00 00:30
9DE0:00 00 00 00 00 00 00 00:7D
9DE8:00 A2 00 00 00 2A 00 00:C5
9DF0:00 00 00 00 00 00 00 00:8D
9DF8:00 A2 00 00 00 2A 15 00:68
9E00:00 00 00 00 00 00 00 00:9E
9E08:00 00 00 A2 00 2A 00 00:2A
9E10:00 00 00 00 00 00 00 00:AE
9E18:00 00 00 A2 00 2A 15 00:CD
9E20:00 88 00 88 14 00 00 00:52
9E28:00 00 00 00 00 00 00 00:C6
9E30:00 88 44 44 14 14 00 14:36
9E38:00 A2 00 00 00 2A 00 00:16
```

```
9E40:00 88 00 88 00 28 00 28:3E
9E48:00 A2 00 00 00 2A 00 00:26
9E50:44 CC 44 CC 14 14 14 14:CE
9E58:51 F3 51 51 15 15 00 00:4B
9E60:44 CC 44 CC 14 14 14 14:DE
9E68:51 A2 51 51 15 3F 00 00:B5
9E70:44 CC 44 CC 14 00 14 00:D6
9E78:51 00 51 00 15 3F 00 00:3D
9E80:44 88 44 CC 14 14 14 14:76
9E88:51 51 51 51 15 3F 00 00:33
9E90:44 CC 44 CC 14 00 14 00:F6
9E98:51 A2 51 00 15 3F 00 00:A1
9EA0:44 CC 44 CC 14 00 14 00:06
9EA8:51 A2 51 00 15 00 00 00:37
9EB0:44 CC 44 CC 14 14 14 00:8E
9EB8:51 00 51 51 15 3F 00 00:C1
9EC0:44 44 44 44 14 14 14 14:0E
9EC8:51 F3 51 51 15 15 00 00:BB
9ED0:44 CC 00 88 00 28 00 28:9A
9ED8:00 A2 00 A2 15 3F 00 00:25
9EE0:44 CC 00 88 00 28 00 28:AA
9EE8:00 A2 00 A2 15 2A 00 00:B7
9EF0:44 44 44 44 14 14 14 28:DE
9EF8:51 A2 51 51 15 15 00 00:49
9F00:44 00 44 00 14 00 14 00:9F
9F08:51 00 51 00 15 3F 00 00:CE
9F10:44 44 44 CC 14 3C 14 14:6F
9F18:51 51 51 51 15 15 00 00:C8
9F20:44 44 44 44 14 3C 14 3C:9F
9F28:51 F3 51 51 15 15 00 00:1C
9F30:44 CC 44 44 14 14 14 14:8F
9F38:51 51 51 51 15 3F 00 00:E4
9F40:44 CC 44 CC 14 14 14 14:BF
9F48:51 F3 51 00 15 00 00 00:7A
9F50:44 CC 44 CC 14 14 14 14:CF
9F58:51 51 51 51 15 3F 00 2A:54
```

```
9F60:44 CC 44 CC 14 14 14 14:DF
9F68:51 A2 51 51 15 15 00 00:BA
9F70:44 CC 44 CC 14 00 14 00:D7
9F78:51 F3 00 51 15 3F 00 00:75
9F80:44 CC 44 CC 00 28 00 28:27
9F88:00 A2 00 A2 00 2A 00 00:EF
9F90:44 44 44 44 14 14 14 14:DF
9F98:51 51 51 51 15 3F 00 00:44
9FA0:44 44 44 44 14 14 14 14:EF
9FA8:51 51 51 51 00 2A 00 00:6D
9FB0:44 44 44 44 14 14 14 14:FF
9FB8:51 F3 51 F3 15 15 00 00:34
9FC0:44 44 44 44 14 14 00 28:23
9FC8:00 A2 51 51 15 15 00 00:C9
9FD0:44 44 44 44 14 14 00 28:33
9FD8:00 A2 00 A2 00 2A 00 00:3F
9FE0:44 CC 00 44 00 14 00 28:23
9FE8:00 A2 51 00 15 3F 00 00:A1
9FF0:00 00 00 00 00 00 00 00:8F
9FF8:00 00 00 00 00 00 00 00:97
A000:7C C6 08 67 D0 D5 11 50:61
A008:C0 19 D1 C9 00 DD 5E 00:F1
A010:DD 56 01 DD 6E 02 DD 66:1D
A018:03 06 10 C5 E5 06 04 1A:94
A020:AE 77 13 23 10 F9 E1 CD:D6
A028:00 A0 C1 10 EE C9 00 00:E7
A030:00 00 DD 6E 00 DD 66 01:1F
A038:06 07 C5 E5 06 06 AF 77:92
A040:23 10 FC E1 CD 00 A0 C1:04
A048:10 F0 C9 DD 6E 00 DD 66:08
A050:01 DD 5E 02 DD 56 03 06:67
A058:20 C5 E5 D5 01 06 00 ED:36
A060:B0 E1 CD 00 A0 E5 D1 E1:16
A068:CD 00 A0 C1 10 EB 3E 20:3D
A070:32 39 A0 CD 32 A0 3E 07:6C
A078:32 39 A0 C9 00 00 00 00:C0
```

Les modes d'adressage (suite)

AUTO-FORMEZ-VOUS A L'ASSEMBLEUR 68000 (5e partie)

L'adressage direct étant d'ores et déjà connu de tous (Micro-Mag n°12), passons au niveau supérieur, j'ai nommé l'adressage absolu long et court.

Mais avant cela, il apparaît nécessaire de définir ce qu'est une adresse effective.

L'adresse effective

On utilise cette expression pour désigner les opérandes utilisés dans les instructions 68000. C'est ainsi qu'une adresse effective englobe tous les types d'opérandes connus. Voyez le tableau ci-contre résumant les 15 différentes adresses effectives.
On utilise couramment le symbole <ae> pour caractériser une adresse effective. Une instruction MOVE peut dès lors s'écrire:

MOVE.f <ae>,<ae>

Une adresse effective sera donc un registre quelconque du 68000 ou un emplacement mémoire directement, voire indirectement défini. Reste à découvrir ce que signifient les autres opérandes.

Symbole	Mode	Symbole	Mode	Symbole	Mode
Dn	Direct	(An)	Indirect	d16 (PC)	Indirect
An	Direct	(An) +	Indirect	d8 (PC, Rn)	Indirect
(x).L	Absolu	- (An)	Indirect	SR	Immédiat
(x).W	Absolu	d16 (An)	Indirect	CCR	Immédiat
#x	Immédiat	d8 (An, Rn)	Indirect	USP	Direct

Les types d'adresses effectives

R.A.M.

$20000	00	00	$20001
$20002	00	00	$20003
$20004	00	00	$20005
$20006	00	00	$20007

Avant FFFFFFFF D0
Après FFFFFF00 D0

Figure 1

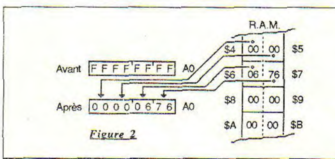


Figure 2

L'adressage absolu, long et court

Nous allons faire intervenir la mémoire dans nos propos. En effet, une adresse absolue (valeur 32 bits) est tout simplement l'une des cases de nos mémoires. Ce n'est donc plus un transfert registre-registre qui a lieu, mais bien un échange entre la mémoire interne du 68000 qui contient les registres, et les mémoires externes telles que la RAM et la ROM. Voyons maintenant le format d'une instruction MOVE correspondante :
MOVE.f (x).f,<ae>

Opération effectuée:
(x).f -> <ae>

Ça se corse ! Examinons un exemple concret (figure 1):

MOVE.B $20000.L,D0

Le (x) correspond, comme dit précédemment, à une valeur 16 ou 32 bits, tandis que f correspond au format de cette adresse. L'intérêt de ce f est simple: pourquoi aurions-nous besoin de 32 bits pour désigner une valeur? 16 bits suffisent largement. Suivant la grandeur de l'adresse, il paraît normal de pouvoir choisir le format de celle-ci; les conséquences seront un gain de vitesse et de mémoire. A ceci, nous indiquerons que le format f consiste en 2 différentes valeurs. Un W indique une adresse codée sur 16 bits et un L une adresse sur 32 bits. Il est donc possible d'adresser la mémoire de $F80000 à $007FFF en utilisant le format 16 bits (adressage absolu court). Si le format est de type long (sur 32 bits), alors l'adresse n'est pas signée et on accède ainsi à la totalité de la mémoire ($0 à $FFFFFF). Une astuce: si l'adresse est codée sur 32 bits, le .L peut être omis. Le codage d'une adresse sur 8 bits n'existe pas, donc pas de MOVE.L $6.B,D0. Voici un exemple d'utilisation d'une adresse 16 bits:

MOVEA.L $4.W,A0 (figure 2)

Quelques explications à propos de ce mode d'adressage: Rappelez-vous que chaque adresse d'une mémoire externe contient un octet. Le 68000 commence par déterminer

l'adresse effective de l'opérande de source, ici une adresse absolue, puis celle de l'opérande de destination, un registre d'adresses. Le 68000 fait ensuite un accès mémoire et récupère, à l'adresse absolue indiquée, le nombre d'octets défini dans le format f de l'instruction MOVE. Enfin, il dépose ces valeurs dans le registre d'adresses.
A ce stade, précisons qu'il n'est pas permis d'utiliser un format autre que 8 bits pour l'instruction MOVE, si l'adresse absolue de l'opérande source et/ou destination est impaire. Ainsi, MOVE.W $20001.L, D0 est impossible. En d'autres termes, on accède à une adresse impaire si (et seulement si) l'accès se fait seulement sur 8 bits comme MOVE.B $455.W,D.
D'un autre côté, on peut évidemment utiliser l'adressage absolu avec un opérande de destination ou même avec les 2 opérandes. Voici 2 exemples.

MOVE.W D1,$1000.W
MOVE.B $40000,$40001
(figures 3 et 4)

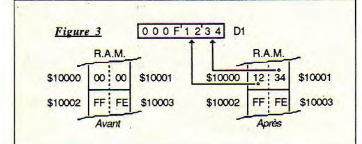


Figure 3

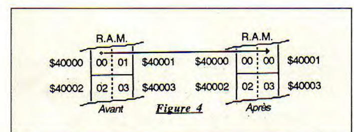


Figure 4

Faites un petit programme de test (voir article précédent) et amusez-vous. N'omettez pas le RTS final après vos MOVEs, sous peine... d'un plantage. Remarquez quand même que le «bas» de la RAM - espace mémoire allant de $0000 à $03FF - est rempli de valeurs système telles les vectorisations pour les exceptions, les interruptions, etc. Une promenade de santé dans les antres du 68000 serait donc imprudente, tout du moins pour l'instant...
Il eut été judicieux de comparer les temps d'exécution avec les modes d'adressage utilisés, mais réservons cela pour plus tard. Il faut auparavant maîtriser l'ensemble des modes et instructions du 68000.

Le mode d'adressage immédiat

L'adressage immédiat est caractérisé par le fait que l'opérande source est une valeur statique, fixée une fois pour toute à la suite du code opératoire de l'instruction utilisée.

Instruction: MOVE.f #x,<ae>
Opération effectuée:
#x -> <ae>

L'opérande source au format immédiat est stocké à l'adresse effective précisée par l'opérande destination. A noter le symbole «#» pour distinguer le mode immédiat du mode absolu abordé précédemment. Voici un exemple:

MOVE.B #$40,D0
(figure 5)

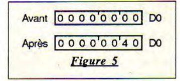


Figure 5

Les 3 formats B,W et L sont possibles. Si le format W est utilisé et si l'opérande destination est un registre d'adresses, alors une extension de signe 32 bits sera effectuée. Le format B est interdit avec un registre d'adresse. Les valeurs de départ des registres et adresses dans nos exemples sont purement fortuites, elles servent juste à bien préciser le changement d'état. Testons ceci avec:

MOVEA.W #$F240,A0
(figure 6)

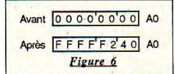


Figure 6

Voyons également, à l'aide de 2 exemples, ce que l'on obtiendrait en utilisant un nombre négatif:

MOVE.W #-$5000,D0 et
MOVE.L #-2,D1
(figure 7 et 8)

Avant 0 0 0 0'0 0'0 0 D0
Après 0 0 0 0'B 0'0 0 D0
Rappel: - $5000 = $B000
Figure 7

Avant 0 0 0 0'0 0'0 0 D1
Après F F F F'F F'F E D1
Rappel: - $2 = $FFFF FFFE
Figure 8

Dernière chose: il est bien sûr inutile d'essayer de mettre une adresse effective de type immédiat pour l'opérande destination. Ceci donnerait: MOVE.f #x,#y ce qui, vous en conviendrez, n'a aucun sens. Le mode d'adressage immédiat d'un opérande destination est donc absolument prohibé sous peine d'une mémorable exception «Illegal Instruction», et cela quelle que soit l'adresse effective de l'opérande source (exception faite des instructions LINK et BTST en mode dynamique.)

• Les flags

Représentés par l'octet utilisateur (CCR) du registre SR, ils sont en constante «mouvance». Ainsi, lorsque l'opérande source est nul, le flag Z (Zéro) sera positionné. Si l'opérande est négatif, le flag N sera mis à 1. On peut donc, sans même faire de test de comparaison supplémentaire, effectuer un branchement.

• L'adressage immédiat accéléré

Il existe en effet une variante du mode immédiat. Il s'agit d'utiliser une instruction de type MOVEQ (contraction de MOVE-Quick). Dans ce cas, l'opérande source est obligatoirement immédiat et l'opérande destination un registre de données. La valeur source est un nombre compris entre -128 et +127, le 68000 effectue une extension 32 bits avant de stocker la valeur finale dans Dn. L'intérêt de cette instruction semble limité au premier abord, mais elle met très peu de temps à être exécutée et l'opérande source est stocké dans le code opératoire même de l'instruction. Voici un exemple de comparaison : MOVEQ #120,D0 et MOVE.L #120,D0 donnent un résultat rigoureusement identique.
1- l'instruction MOVEQ occupe 2 octets en RAM, et demande 4 cycles d'exécution.
2- l'instruction MOVE.L, elle, occupe 6 octets, et 12 cycles pour son exécution.

L'important chez un programmeur étant l'optimisation (tant en vitesse qu'en mémoire) de son programme, nous vous incitons plus que jamais à choisir au mieux vos instruc-

tions, afin de réaliser un programme tendant vers la perfection, encore très rare de nos jours...

Il n'est pas trop tard pour expliquer ce qu'est un cycle: La fréquence d'un 68000 classique est 7.16 Mhz, c'est-à-dire qu'il se produit 7160000 fronts d'horloge par seconde. Ces fronts étant obtenus à l'aide d'un oscillateur ainsi que d'un circuit externe de décodage. L'exécution des instructions est fixé par rapport à un nombre défini de cycles. Vous imaginez facilement que plus le 68000 a d'accès mémoire à faire, plus le nombre de cycles sera élevé. Si, en revanche, il a juste quelques opérations à réaliser avec ses registres, alors la vitesse sera d'autant plus rapide. En moyenne, une instruction 68000 met 20 cycles à être exécutée, le 68000 peut donc exécuter 358000 instructions par seconde. Et encore, il est gêné en temps normal par le DMA qui lui «pique» pas mal de cycles dans ses accès à la RAM. Une adresse effective, à elle seule, peut occuper jusqu'à 16 cycles (sans considérer le DMA), alors que l'utilisation de registres internes amène un temps d'exécution presque

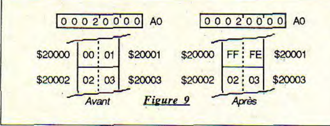


Figure 9

immédiat. Il est de ce fait déconseillé d'utiliser à tort et à travers l'adressage absolu. On lui préfèrera en effet l'adressage indirect, plus rapide et plus court en mémoire.

Mode d'adressage indirect de registre d'adresses

La spécification «indirect» désigne en fait le caractère dynamique de l'adressage. Une fois l'instruction tapée, quelle que soit l'adresse où elle se trouve logée, son exécution sera possible sans modification préalable. D'où une puissance du 68000 qui ne cessera de s'accroître, vous le verrez !

Instruction générale:
MOVE.f <ae>,(An)
Opération effectuée:
<ae> -> (An)

Ah, du nouveau sur le front de l'adressage, nous apercevons effectivement que l'opérande destination est «entre parenthèses». C'est en fait pour une raison simple: A quoi sert un registre d'adresses? A adresser!... Bien! Il contient donc une valeur 32 bits indiquant l'adresse où le 68000 ira faire son travail. Cette instruction est radicalement différente du MOVE.f <ae>,An. Cette dernière charge dans An une certaine valeur calculée à partir de l'adresse effective de l'opérande source. Avec l'adressage indirect, ce n'est pas An qui sera modifié, mais l'emplacement mémoire sur lequel il pointe. Pour clarifier ceci, voici un exemple concret.

MOVE.L #$20000,A0 ; -> A0=$20000
MOVE.W #$FFFE,(A0) ; -> $20000 <- $FF, $20001 <- $FE (figure 9)

Notez au passage que ceci ne modifie en rien l'état de An. Le flag N sera positionné suite à ces 2 instructions.

Maintenant que nous connaissons l'adressage absolu, nous pourrions effectuer la même opération que la précédente : MOVE.W #$FFFE,$20000.L (notez que taper MOVE #-

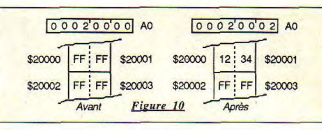


Figure 10

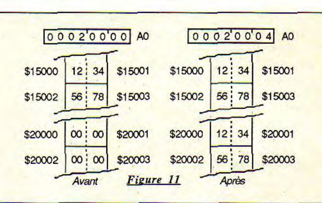


Figure 11

2,$20000 reviendrait au même). Avec l'adressage indirect, le 68000 occupe 24 cycles et seulement 20 avec l'adressage absolu.

Donc, c'est en contradiction avec ce qui a été dit plus haut, mais il faut savoir que dans le 1er cas, le registre A0 est conservé et utilisable pour un prochain accès, ce qui est différent du second cas où aucun registre n'est initialisé.

Adressage indirect de registre d'adresses avec postincrémentation

Instruction MOVE:
MOVE.f <ae>,(An)+

Opérations effectuées:
<ae> -> (An)
An <- An+f

Ici, non seulement An sert de pointeur sur l'adresse mémoire à affecter, mais en plus, APRES l'opération (d'où postincrémentation), An se verra augmenté d'une certaine valeur calculée en fonction du format de l'opérande source. L'opérande source ne sera pas affecté. 2 exemples :

MOVE.W #$1234,(A0)+ ; format W donc A0 <- A0+2
MOVE.L $15000,(A0)+ ; format L donc A0 <- A0+4
(figures 10 et 11)

Ce type d'adressage est fortement utilisé lorsqu'il s'agit de «boucles», par exemple :

```
      MOVE.L #$30000,A0   ; A0=$30000
      MOVEQ #10,D0        ; 10+1 fois
LOOP: MOVE.B D0,(A0)+     ; transfert
      DBF D0,LOOP         ; bouclage
      RTS                 ; retour éditeur
```

Suite à ce programme, le contenu en mémoire sera le suivant :

$30000	->	10
$30001	->	9
$30002	->	8
$30003	->	7
$30004	->	6
$30005	->	5
$30006	->	4
$30007	->	3
$30008	->	2
$30009	->	1
$3000A	->	0

D0 contiendra la valeur $FFFFFFFF (-1), et A0 $3000B.

Là encore, la dynamique des instructions de notre bon vieux 68000 transforme la programmation en un véritable jeu d'enfant...

Adressage indirect de registre d'adresses avec prédécrémentation

Inutile de faire tout un discours sur ce mode d'adressage. Le précédent permettait une POSTincrémentation, celui-ci effectue une PRE-décrémentation. C'est à dire que An sera d'abord décrémenté puis le transfert aura lieu.

Instruction MOVE :
MOVE.f <ae>,-(An)
Opérations effectuées:
An <- An-f
<ae> -> (An)

Un exemple:
MOVE.L #$40000,A0
MOVE.W #$5678,-(A0)
(figure 12)
Quelques remarques à propos de l'adressage indirect de registre d'adresses:

• Il n'est pas permis de faire de même avec des registres de données.
• L'inc/décrémentation est réelle, le résultat interne de l'opération effectuée par le 68000 est ensuite chargé dans An.
• Aucune extension de signe n'est effectuée.
• Les 3 formats B,W et L sont autorisés.

Bien sûr, pour nos explications, seul l'opérande destination s'est vu affecté d'une adresse effective indirecte. Il est évident que cela peut également s'appliquer dans d'autres cas. De plus, là aussi, les 3 formats sont possibles.
Avant de se quitter, examinons un dernier exemple (ci-contre) qui correspond à un programme de copie de zones mémoires. Tous les types d'adressage indirects sont utilisés.
Quelques modes d'adressage restent à voir prochainement. En attendant, assemblez-bien !

Stéphane Rodriguez

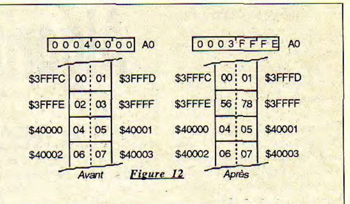


Figure 12

```
      MOVE.L #$20000,A0  ; 1ère adresse de départ.
      MOVE.L #$22000,A1  ; 2ème adresse de départ.
      MOVE.L #$30000,A2  ; 3ème adresse de départ.
      MOVE.W #$1000,D0   ; Nombre de boucles
LOOP: MOVE.B (A0)+,(A1)  ; 1ère copie (Source postincrémentée)
      MOVE.B -(A1),D1    ; 2ème copie (Source prédécrémentée)
      NOT.L D1           ; Inversement de D1
      MOVE.B D1,(A2)+    ; Copie en A2 (Destination postincrémentée)
      NOT.L D1           ; Ré-inversement.
      DBF D0,LOOP        ; Bouclage
      RTS                ; Retour
```

Algorithmique

LA RECURSIVITÉ

Voisine de la récurrence mathématique, la récursivité n'est pas seulement une technique de programmation. C'est aussi une méthode de réflexion qui permet d'aborder des problèmes a priori insolubles comme ceux posés par l'intelligence artificielle.

Actuellement, la plupart des langages évolués C, Pascal, Lisp, certains Basic,... etc acceptent des fonctions récursives et proposent ainsi une facilité dont il serait dommage de se passer. L'idée essentielle consiste à transformer un problème complexe en plusieurs sous-problèmes SIMILAIRES mais plus simples et AINSI DE SUITE jusqu'au moment ou on est ramené à des cas suffisamment simples pour être traités directement. Exprimé de façon guerrière, ceci s'appelle «Diviser pour règner». Voyons les premières conséquences:

• Il faut trouver le moyen de décomposer le problème, ce n'est pas toujours facile.

• Il est impératif de s'assurer que les sous-problèmes sont calculables, c'est-à-dire qu'on aboutira après un certain nombre d'étapes à un cas trivial. Sinon, soit on a affaire à un cercle vicieux, soit le nombre des sous-problèmes se met à croître vertigineusement et finit par faire planter l'ordinateur.

Premiers pas

Voici quelques exemples illustrant cette manière de penser:

• Monter un escalier de *n* marches se remène à :

1- Si l'escalier ne comporte qu'une marche, on monte d'une marche et c'est finit (le cas simple).

2- Sinon, il suffit de monter d'une marche (cas simple) et de recommencer l'algorithme avec un escalier de n-1 marches (sous-problème similaire).

• Donner la liste des ancêtres d'une personne (en négligeant les collatéraux) se ramène à :

1 - Si on connaît le père.

a) Afficher le nom du père (on sait faire).

b) Recommencer avec la liste des ancêtres du père (problème similaire).

2 - Si on connaît la mère.

a) Afficher le nom de la mère (on sait faire).

b) Recommencer avec la liste des ancêtres de la mère (problème similaire).

3) - Si on ne connaît ni père, ni mère, c'est finit (cas simple).

• Parfois, on n'est pas sur d'atteindre un cas simple, il faut alors une condition d'arrêt pour échapper à une régression infinie ou dépassant les capacités de l'ordinateur ou encore, qui serait inacceptable en temps machine. Ce problème se rencontre (même pour un humain) au cours d'une partie d'échecs, de dames, etc... Pour chaque déplacement envisagé, il faut évaluer la position, imaginer la réponse de l'adversaire et...ainsi de suite jusqu'à une profondeur donnée (L'algorithme classique dans les problèmes à deux joueurs s'appelle le minimax, car l'un des joueurs tente de minimiser le gain espéré alors que l'autre le maximise).

• On peut encore citer de nombreux exemples où la notion de récursivité est omniprésente, comme les systèmes experts, l'analyse syntaxique, le calcul symbolique traité par ordinateur, etc. En résumé, l'énoncé d'un problème dans lequel on trouve l'expression «...et ainsi de suite jusqu'à ce que telle condition soit remplie» est de nature essentiellement récursive (La boucle do...while en est d'ailleurs le cas le plus simple). Voyons maintenant comment traiter quelques cas pratiques simples...

L'exemple «bateau» est celui de la fonction factorielle. On sait qu'elle est définie par :
fact(n) = n! = n*(n-1)*(n-2)*...* 1
et on peut la calculer à l'aide de la fonction suivante:

```
fact( n )
int n ;
{
  int f = n ;
  while( --n > 0 )
    f = f * n ;
  return( f ) ;
}
```

N. B. : Ici, comme dans la suite et pour éviter les problèmes d'intendance, on considérera qu'on se borne à des valeurs de n suffisamment petites pour ne pas devoir faire intervenir les entiers longs. Mais on peut essayer une approche récursive en remarquant que:
fact(1) = 1 et
fact(n) = n * fact(n-1)
Ayant lu attentivement ce qui précède et sachant en particulier que le C est récursif, on tape avec confiance :

```
fact( n )
int n ;
{
  if( n<= 1 )
    return( 1 ) ;
  else
    return( n * fact( n-1 )) ;
}
```

Et ca marche! Examinons un cas similaire mais un chouia plus compliqué. Soit la fonction donnée par:

f(1) = f(2) = 1 et
f(n) = f(n-1) + f(n-2) si n > 2

La solution récursive s'écrit immédiatement :

```
f( n )
int n ;
{
  if( n < 3 )
   return( 1 ) ;
```

```
else
  return( f( n-1 ) + f( n-2 )) ;
}
```

Cet exemple est encore relativement facile à écrire de façon non récursive. Il suffit de calculer les valeurs de 1 à n en gardant en mémoire les deux dernières, dans un tableau ou dans des variables (f(n) donne les valeurs successives de la suite dite «de Fibonacci» et qui sont: 1 1 2 3 5 8 13....). Mais si vous cherchez à trouver une solution non récursive à la prochaine, vous allez jongler...

```
ack( m, n )
int m, n ; /* de petites valeurs
*/
{
  if( m==0 )
    return( n+1 ) ;
  else if( n==0 )
    return( ack( m-1, 1 )) ;
  else
    return( ack( m-1, ack( m, n-
1 ))) ;
}
```

Listing 1

Désassemblage de la fonction fibo() générée par le compilateur du Mégamax C...

```
fibo:/* global */
        LINK    A6,L$6      /* Pas de variables locales => L$6 = 0 */
        CMPI    #2,8(A6)    /* 8(A6) contient n */
        BGT     L8(PC)      /* Si n > 2, on va en L8 */
        MOVE    #1,D0       /* La valeur de retour est mise dans D0 */
        BRA     L7(PC)      /* Et on sort */
L8:
        MOVE    8(A6),D0    /* On récupère n */
        SUBQ    #1,D0       /* D0 = n-1 */
        MOVE    D0,-(A7)    /* Passage de n-1 par la pile */
        JSR     fibo(PC)    /* Exécute fibo( n-1 ) */
        ADDQ.L  #2,A7       /* Restauration de la pile */
        MOVE.L  D0,-(A7)    /* Empile le résultat */

        MOVE    8(A6),D0
        SUBQ    #2,D0
        MOVE    D0,-(A7)
        JSR     fibo(PC)
        ADDQ.L  #2,A7       /* fibo( n-2 ) est dans D0 */

        MOVE.L  D0,D1       /* Passage dans D1 */
        MOVE.L  (A7)+,D0    /* Récupération de fibo( n-1 ) */
        ADD     D1,D0       /* D0 = fibo( n-1 ) + fibo( n-2 ) */
        BRA     L7(PC)      /* Kolossale finesse du compilo */
L7:
        UNLK    A6          /* Restauration de la pile */
        RTS                 /* Retour */
L$6:    .EQU    #0
```

Comment ça marche?

Euh...Comme si vous le faisiez vous-même... Reprenons la factorielle. Pour calculer sa valeur avec n = 5, on fait :

fact(5) = 5*(fact(4)) = 5*(4 *(fact(3))) et en fin de compte :
fact(5) = 5*(4*(3* (2*(1))))

Appelons «incarnation», l'ensemble généré par l'appel d'une fonction. C'est la zone - au sommet de la pile - contenant paramètres et variables locales. Elle est détruite en fin d'exécution et la pile est restaurée après que la procédure appelante aie récupéré le résultat. Le calcul de fact(n) va donc commencer par l'ouverture de n-1 incarnations successives avant de pouvoir renvoyer une valeur qui sera celle de fact(1) = 1. C'est seulement alors, que toutes ces incarnations vont se refermer en cascade en effectuant les multiplications. On peut dire qu'on a affaire à une remontée de l'information vers les couches les plus hautes ou, en utilisant le langage des arbres, que la valeur du noeud terminal - et des autres - est envoyée à la racine. Concrétisons les choses (si possible)... Imaginons 5 poupées chinoises emboîtées dont la plus interne a la valeur 1 écrite dessus. Sur le couvercle de la première, on écrit :
fact(5) = 5 * ... = ...
sans savoir pour le moment par quoi remplacer les «...». Puis on retire soigneusement le couvercle pour faire apparaître la suivante sur laquelle on écrit :
fact(4) = 4 * ... = ...
Et ainsi de suite :
fact(3) = 3 * ... = ...
fact(2) = 2 * ... = ...
fact(1) = 1
Ayant obtenu l'information nécessaire, il ne reste plus qu'à remettre les couvercles dans le bon ordre en remplissant les «...». Ce qui donne la suite d'opérations :

fact(2) = 2 * 1 = 2
fact(3) = 3 * 2 = 6
fact(4) = 4 * 6 = 24
fact(5) = 5 * 24 = 120

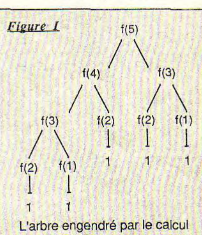
Bien entendu, le cas de la factorielle est particulièrement simple et se résoud à l'aide d'une seule série de poupées. Mais pensez au calcul de la suite de Fibonacci... Les fanatiques de l'assembleur en trouveront un désassemblage commenté dans l'encadré 1, montrant comment un compilateur peut réagir face à la récursivité.

Tout n'est pas rose

Si la mise en oeuvre de la récursivité se révèle souvent nécessaire, puissante et facile à coder, on ne peut passer sous silence les problèmes qu'elle entraîne et qui doivent inciter à une utilisation prudente. Nous avons vu qu'un appel récursif se traduisait par une réservation d'espace sur la pile. Mais, outre la mémoire, il faut aussi penser que cette opération demande un temps certain. On peut en conclure qu'un algorithme récursif n'est pas, en général, le moyen le plus efficace d'arriver au but. Penchons nous un peu plus sur la suite de Fibonacci. De la relation :

f(n) = f(n-1) + f(n-2),
on tire en développant :
f(n) = [f(n-2) + f(n-3)] + [f(n-3) + f(n-4)]

et le fameux «ainsi de suite»...
Deux remarques s'imposent :
- Il va falloir - inutilement! - recalculer plusieurs fois les mêmes valeurs (voir figure 1).
- Comme chaque calcul se subdivise en deux autres, la complexité de l'algorithme va être en $2^{(n-2)}$. C'est-à-dire que le temps d'exécution va croître exponentiellement en fonction de n. Inacceptable, alors qu'on dispose d'un algorithme itératif simple qui est en O(n) et que voici :



L'arbre engendré par le calcul de fibo(5). Le parcours se fait en préordre.

```
fibo( n )
int n ;
{
  int u1 = 1, u2 = 1, tampon, i ;
  if( n < 3 )
    return( 1 ) ;
  for( i = 3 ; i <= n ; i++ )
  {
    tampon = u2 ;
    u2 += u1
    u1  = tampon ;
  }
  return( u2 ) ;
}
```

Moralité, ne pas sauter sur la première solution récursive venue! Si on s'aperçoit que la complexité spatiale ou temporelle du programme va poser des problèmes, il est bon d'essayer de trouver une solution simplement itérative... si elle existe. Une question se pose: Peut-on toujours éliminer la récursivité? Disons qu'il est possible de l'éviter formellement en remplaçant l'appel récursif - et la sauvegarde de l'environnement dans la pile - par la sauvegarde des données importantes dans un ou plusieurs tableaux sur lesquels se promènent des pointeurs, correspondant au niveau d'imbrication atteint. Cette technique revient en fait à gérer manuellement la récursivité, comme l'ont fait des générations de programmeurs avec des langages ne leur proposant pas ce cadeau. Même combat pour celui qui ne s'est pas rendu compte de la récursivité sous-jacente de son problème et se retrouve dans la situation de M. Jourdain... Il n'est pas évident que ce genre de solution fasse gagner beaucoup de temps à l'exécution, bien qu'on puisse ici sélectionner les variables à sauver. Quant au coût du développement... On ne peut donc vraiment parler d'élimination de récursivité que lorsqu'on a fait disparaître un ou plusieurs de ces tableaux - ou piles -. Les moyens (excepté parfois... le bon sens) qui le permettent ne sont absolument pas immédiats et je préfère renvoyer à la bibliographie en fin d'article, où sont examinés à fond certains exemples non triviaux. Concluons avec quelques graphismes un peu plus «bandants» que la suite de Fibonacci et autres factorielles.

```
#define clearw() v_clrwk( handle )
#define closew() v_clsvwk( handle )
#define showcur() v_show_c( handle,0 )
#define hidecur() v_hide_c( handle )
#define HAUTEUR 199

int
        contrl[12], intin[128], ptsin[128], intout[128], ptsout[128];
int
        handle, work_out[57], work_in[12];
int
        v_offx, v_offy, v_graphmat[10] ;

openw()
{
        int i ;
        for ( i = 0 ; i < 10 ; work_in[ i++ ] = 1 ) ;
        work_in[10] = 2 ;
        v_opnvwk( work_in, &handle, work_out );
        vsf_perimeter( handle, 1 );
        vsf_interior( handle, 2 );
        vsf_color( handle, 1 );
        vswr_mode( handle, 1 );
        vsf_style( handle, 1 );
}

setcolor( x )
int x ;
{
        x &= 3 ;
        vsm_color( handle,x ) ;
        vsl_color( handle,x ) ;
}

origin( x, y )
int x, y;
{
        v_offx = x ;
        v_offy = y ;

        move( x, y ) ;
}


move( x, y )
int x, y;
{
        register int *ptr = v_graphmat;
        *ptr++ = x + v_offx ;
        *ptr = HAUTEUR -(y+v_offy);
}

draw( x, y )
int x, y;
{
        register int *ptr = v_graphmat + 2;

        *ptr++ = x;
        *ptr  = HAUTEUR -(y+v_offy) ;
        v_pline( handle, 2, v_graphmat );
        move( x, y );
}

plot( x, y )
int x, y ;
{
        register int *ptr = v_graphmat + 2 ;

        *ptr++  = x + v_offx;
        *ptr–  = HAUTEUR - (y+v_offy) ;
        v_pmarker( handle, 1, ptr );
}
```

Listing 2

Le fichier <graphset.h>

Des dessins récursifs

Les programmes proposés ont été écrits pour tourner sur ST en moyenne résolution (620x200). Il utilisent tous le fichier header <graphset.h>, qui contient quelques fonctions graphiques rudimentaires obtenues à partir des appels VDI. Pour ceux qui voudraient s'en passer ou utiliser des moyens plus rapides comme les LINEA, voici le détail des routines :

- Le système des coordonnées est modifié, l'origine est en bas à gauche et les «y» croissent vers le haut.
- openw() ouvre une station de travail qu'il FAUT fermer à la fin avec :
- closew()
- clearw() efface l'écran.
- hidecur() cache la souris.
- showcur() montre la souris.
- origin(x, y) déplace l'origine de l'écran au point (x, y) et permet d'utiliser des coordonnées négatives.
- setcolor(c) fixe la couleur des points et droites.
- plot(x, y) affiche un point relativement au repère.
- move(x, y) positionne le pointeur graphique relativement au repère.
- draw(x, y) trace une droite entre le pointeur graphique (x0,y0) et le point (x, y).

Une cercle bizarre

Le programme en encadré 3 s'appelle cercle. c. L'idée est de tracer un cercle de rayon r mais dont chaque point serait lui-même le centre d'un cercle de rayon inférieur. Et ainsi de suite... Pour aller suffisamment vite, il faut travailler en entiers et précalculer un tableau de sinus et de cosinus pour chaque point dont le nombre est fixé à 35 par cercle. Les valeurs trigonométriques sont multipliées par 256 afin de ne pas trop perdre de précision et on redivise ensuite pour le résultat final. Les appels récursifs cessent quand le rayon devient inférieur à 25,

c'est la condition d'arrêt. On remarque que les ordonnées sont divisées par 2 avant d'afficher un point. C'est pour corriger le phénomène d'étirement vers haut dû à la moyenne résolution (640x200). Il est facile de construire des variations sur ce thème : Par exemple, en remplaçant les cercles par des carrés, etc.

Listing 3
Cercles.c

```
#include <math.h>
#include <osbind.h>
#include <graphset.h>

#define pi2 6.28
#define NB  35

int si[NB+5], co[NB+5];

main()
{
   double k, pas;
   int i=0;

   openw() ;
   clearw();
   origin( 320, 100 ); /* centre de l'écran */
   hidecur();
   pas= pi2 / ( double ) NB;
   for( k = 0 ; k < pi2 ; k += pas )
   {
        si[i]  =( int )( 256.0*sin( k ) );
        co[i++]=( int )( 256.0*cos( k ) );
   }
   setcolor( 2 ) ;
   circle( 0, 0, 80 );
   Crawcin();
   showcur();
   closew() ;
}

circle( x, y, r )
int x, y, r;
{
   register int i, dx, dy, dr;
   register int *ptrco, *ptrsi;

   plot( x, ( y >> 1  ) );
   if( r < 25 )
        return ;
   for( ptrco=co, ptrsi = si, i=0 ; i<NB ; i++ )
   {
        dr=( r*100 )/150;
        dx=( r * *ptrco++ ) >> 8;
        dy=( r * *ptrsi++ ) >> 8;
        circle( x+dx, y+dy, dr );
   }
}
```

Le flocon de Von Kock

Cette...pathologie semble surgir des cauchemards d'un mathématicien à la cervelle dévastée par les x. Jugez plutôt :

- La longueur d'une partie quelconque de la courbe est infinie.
- La courbe est partout continue et nulle part dérivable.

Figure 2

(xf, yf)
(kx, ky)
(x2, y2)
60°
(x1, y1)
60°
(x0, y0)

Flocon de Von Kock

- La surface qu'elle englobe est finie.

Pour la construire, on utilise l'algorithme suivant (voir figure 2) :

- On prend un triangle équilatéral qu'on appelle «graine».
- Au centre de chacun des segments de droite ainsi définis, on fait germer un petit triangle équilatéral dont chaque côté fait le tiers du segment initial.
- Et ainsi de suite avec tous les segments de droite présents sur le dessin.
- La condition d'arrêt est donnée par la variable globale lim, qui représente la longueur minimale d'un côté.

Ces types de courbes s'appellent «fractales» et ont été étudiées sérieusement pour la première fois par Benoît Mandelbrojt. Elles permettent de prendre en compte le côté «non euclidien» de certains phénomènes naturels (voir bibliographie). Compte tenu de la résolution d'un ordinateur, on ne peut en donner qu'une approximation gra-

Listing 4
Flocon de Von Kock

```
#include <osbind.h>
#include <graphset.h>

long lim = 200L ;

main()
{
   openw() ;
   origin( 0, 0 ) ;
   do {
        clearw() ;
        flocon() ;
        lim /= 3L ;
        Crawcin() ;
   } while( lim > 0L ) ;
   closew() ;
}

cote( x0, y0, xf, yf ) /* trace l'un des cotés */
int x0, y0, xf, yf ;
{
   int x1, y1, x2, y2, nx, ny, dx, dy ;
   register long kx, ky, si = 113511, co = 65536 ;

   dx = ( xf-x0 )/3 ; /
   dy = ( yf-y0 )/3 ;
   if( dx*dx + dy*dy > lim*lim ) /* pythagore */
   {
        x1 = x0 + dx ; y1 = y0 + dy ;
        x2 = xf - dx ; y2 = yf - dy ;
        kx = ( co*dx - si*dy ) >> 17 ;
        nx = x1 + ( int ) kx ;
        ky = ( si*dx + co*dy ) >> 17 ;
        ny = y1 + ( int ) ky ;
        /* Le sommet du nouveau triangle est trouvé par */
        /* une rotation de 60 degrés */
        cote( x0, y0, x1, y1 ) ;
        cote( x1, y1, nx, ny ) ;
        cote( nx, ny, x2, y2 ) ;
        cote( x2, y2, xf, yf ) ;
        /* Les appels récursifs sur les 4 nouveaux /*
        /* segments */
   }
   else
   {
        move( x0, y0 >> 1 ) ;
        draw( xf, yf >> 1 ) ;
        /* Quand on a atteint la condition d'arret, il ne */
        /* reste plus qu'à tracer  */
   }
}

flocon()
{
   int x0, y0, x1, y1, x2, y2 ;

   x0 = 100 ; y0 = 90 ;
   x1 = 260 ; y1 = 365 ;
   x2 = 420 ; y2 = 90 ;
   cote( x0, y0, x1, y1 ) ;
   cote( x1, y1, x2, y2 ) ;
   cote( x2, y2, x0, y0 ) ;
        /* un flocon comporte 3 côtés */
}
```

phique. Mais cette dernière en montre bien le caractère intrinsèquement récursif, qui n'est pas difficile à mettre en oeuvre quand on en a saisi le principe. Le seul problème consiste à établir les relations mathématiques permettant le passage de l'étape n à l'étape n+1.

La courbe de Péano

La présente courbe se programme de la même manière que le flocon. A partir de la graine donnée (voir figure 3), on cherche les coordonnées des points conduisant à l'étape suivante et ainsi de suite jusqu'à la condition d'arrêt. Notons que ce n'est pas la meilleure façon d'aller de A à B...

Les deux courbes suivantes sont très proches l'une de l'autre par leur construction (figures 4 et 5) qui consiste à décomposer chaque segment de façon simple. Par contre et au lieu de passer les coordonnées début et fin de chaque segment, on se contente d'en donner l'origine et d'indiquer longueur et direction. Au même titre que la courbe de Péano, elles donnent une bien mauvaise réponse au problème du plus court chemin d'un point à un autre...

Vous verrez, en exécutant ces petits programmes, que ces courbes fractales ont de curieuses propriétés. Par exemple, si on prélève une partie et qu'on la grossisse convenablement, on obtient

```
#include <osbind.h>
#include <graphset.h>

long lim = 250L, si = -20L, co = 40L ;

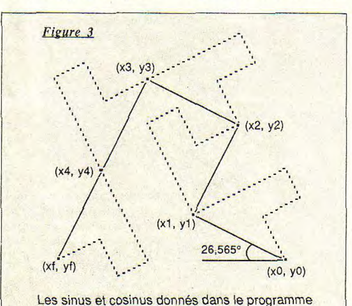
main()
{
    nt i, x0 = 460, y0 = 60, x1 = 180, y1 = 60 ;

    openw() ;
    hidecur() ;
    clearw() ;
    setcolor( 1 ) ;
    origin( 0, 0 ) ;
    do
    {
        peano( x0, y0, x1, y1 ) ;
        lim /= 2L ;
        Crawcin() ;
        clearw() ;
    } while ( lim > 1L ) ;
    showcur() ;
    closew() ;
}

peano( x0, y0, xf, yf )
int x0, y0, xf, yf ;
{
    int v1x, v1y, v2x, v2y, dx, dy, x1, y1, x2, y2, x3, y3, x4, y4 ;
    long kx, ky, ldx, ldy ;

    ldx = ( long ) ( dx = xf - x0 ) ;
    ldy = ( long ) ( dy = yf - y0 ) ;
    if ( ldx*ldx + ldy*ldy > lim*lim )
    {
        kx = ( co*dx - si*dy )/100L ; v1x = ( int ) kx ;
        ky = ( si*dx + co*dy )/100L ; v1y = ( int ) ky ;
        x1 = x0 + v1x ; y1 = y0 + v1y ;
        v2x = v1y ; v2y = -v1x ;
        x2 = x1 + v2x ; y2 = y1 + v2y ;
        x3 = x2 + v1x ; y3 = y2 + v1y ;
        x4 = x1 + v1x ; y4 = y1 + v1y ;
        peano( x0, y0, x1, y1 ) ;
        peano( x2, y2, x1, y1 ) ;
        peano( x2, y2, x3, y3 ) ;
        peano( x3, y3, x4, y4 ) ;
        peano( xf, yf, x4, y4 ) ;
    }
    else {
        move( x0, y0 >> 1 ) ;
        draw( xf, yf >> 1 ) ;
    }
}
```

Listing 5
Courbe de Péano



Les sinus et cosinus donnés dans le programme sont multipliés par 100 $\frac{\sqrt{5}}{5}$.

Courbe de péano

```
#include <osbind.h>
#include <graphset.h>

#define qrt 0.707106781
#define sqr2 1.41421
#define LMIN 2.0

float x0, y0, length ;

float tab[8][2] = {
    { 1.0, 0.0 }, { qrt, qrt }, { 0.0, 1.0 }, { -qrt, qrt },
    { -1.0, 0.0 }, { -qrt, -qrt }, { 0.0, -1.0 }, { qrt, -qrt }
}; /* Tableau de sinus et cosinus pour gagner de temps */

main()
{
    openw();
    origin( 0, 0 ) ;
    length = 128.0 ;
    while( length >= LMIN )
    {
        clearw();
        move( (int) ( x0 = 450.0 ), (int) ( y0 = 150.0 ) );
        Ccurve( 128.0, 4 );
        Crawcin();
        length /= sqr2 ;
    }
    length = 160.0 ;
    while( length >= LMIN )
    {
        clearw();
        move( ( int ) ( x0 = 170.0 ), ( int ) ( y0 = 70.0 ) );
        dragon( 160.0, 0, 1 );
```

(Suite du listing page suivante)

```
        Crawcln();
        length /= sqr2 ;
    }
    closew() ;
}

dragon( len, ang, sign )
float len;
int ang, sign;
{
    float l;
    if ( len <= length )
    {
        ang &= 7;
        if ( ang <0 ) ang += 8;
        polaire( len, ang );
    }
    else
    {
        l=len/sqr2;
        dragon( l, ang+sign, 1 );
        dragon( l, ang-sign, -1 );
    }
}

Ccurve( len, ang )
float len;
int ang;
{
    float l;
    if ( len <= length )
    {
        ang &= 7;
        if ( ang <0 ) ang += 8;
        polaire( len, ang );
    }
    else
    {
        l=len/sqr2;
        Ccurve( l, ang+1 );
        Ccurve( l, ang-1 );
    }
}

    /* Polaire trace un segment d'origine ( x0, y0 ), de longueur */
    /* len et faisant un angle ang avec l'horizontale */

polaire( len, ang )
float len;
int ang;
{
    static int col;

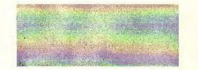
    setcolor( ++col );
    if( col%4 == 3 ) setcolor( col++ );
        /* change de couleur à chaque segment en évitant celle */
        /* du fond */
    x0 += 2*tab[ang][0]*len;
    y0 += tab[ang][1]*len;
    draw( ( int ) x0, ( int ) y0 );
}
```

une courbe semblable au tout. Il est aussi possible de créer de nouveaux monstres en introduisant des valeurs aléaoires ou par tout autre moyen !
ET AINSI DE SUITE, JUSQU'A EPUISEMENT...

Jean-Yves Trétout

Bibliographie

Sur la récursivité :
«Les bases de la programmation» par J. Arsac chez DUNOD.
«Méthodes de programmation» par B. Meyer et C. Baudouin chez EYROLLES.
Au sujet des fractals :
«Les objets fractals» par B. Mendelbrot chez FLAMMARION.
«Fractals: Dimension non-entières et applications» sous la direction de G. Cherbit chez MASSON.



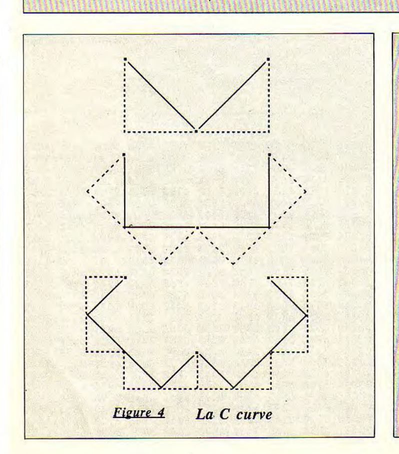


Figure 4 **La C curve**

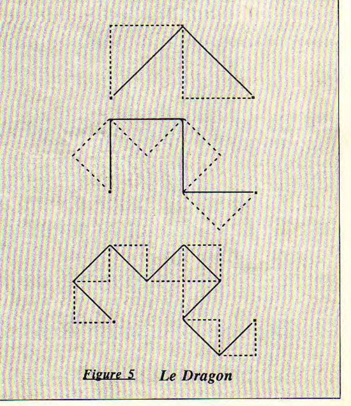


Figure 5 **Le Dragon**

Super copieur

SCOPY

Possesseurs d'un seul drive, soyez heureux! Fini le rituel grille-pain de la copie en plusieurs passes via le Disk RAM.

SCOPY (contraction de Super-Copy) apporte 2 éléments nouveaux dans la manière d'effectuer vos copies.

• Il traite des fichiers de nimporte quelle longueur. Au préalable, il permet d'apprécier la quantité de mémoire restante utilisable à cet effet.

• Il fait l'impasse sur le Disk RAM limité à 200 Ko. La copie s'effectue directement.

Utilisation

SCOPY se divise en 2 parties.

• la partie INFO où l'on accède en tapant SCOPY sous CLI sans y adjoindre le moindre paramètre. Après le copyright traditionnel, s'affiche un nombre hexadécimal précisant la quantité de mémoire CHIP disponible. Si les heureux possesseurs d'au moins 1 méga de RAM n'ont guère l'usage de cet utilitaire, les 512K-istes en revanche, sont vivement intéressés par cette valeur. Tant que la longueur d'un fichier est inférieure à la valeur indiquée, SCOPY est utilisable. Viennent ensuite les précautions à prendre...

• la partie EFFECTIVE, ou fonction copie de SCOPY qui s'obtient par l'ajout d'un paramètre: le nom du fichier que vous désirez copier (Exemple, SCOPY df0:c/SetMap). A ce stade, précisons que le nom du drive utilisé doit obligatoirement être spécifié (ici «DF0:»). Pour le reste, le nom du fichier est on ne peut plus classique et les règles que vous connaissez déjà restent les mêmes. Un message apparaît ensuite vous demandant d'insérer la disquette source. Changez de disquette le cas échéant, puis pressez en connaissance de cause l'un des deux boutons de la souris.

- Le bouton droit sert à sortir du programme en cas de lancement erroné de SCOPY, ou d'erreur dans la saisie du nom du fichier.

- Le bouton gauche avertit l'Amiga que vous êtes prêt. SCOPY charge le fichier puis demande d'insérer la disquette destination (celle où vous désirez copier votre précieux fichier).

Là encore, pressez l'un des 2 boutons après avoir inséré la disquette susdite qui doit contenir suffisamment d'espace libre pour héberger le fichier. La sauvegarde effectuée, retour au CLI comme par enchantement.

Sauvegarde

Les codes hexadécimaux du listing sont à entrer par l'utilitaire *Amiga Saisie* en vous reportant, bien sûr, à son mode d'emploi (Micro-Mag n°10). Longueur en octets à spécifier: 1796.

Stéphane Rodriguez

```
ØØØØ1:ØØØØ Ø3F3 ØØØØ ØØØØ ØØØØ ØØØ2 ØØØØ ØØØØ:Ø3F5
ØØØØ2:ØØØØ ØØØ1 ØØØØ AEEA ØØØØ ØØØ1 ØØØØ Ø3E9:B2D5
ØØØØ3:ØØØØ Ø191 534Ø 67Ø4 6ØØØ ØØ72 2C78 ØØØ4:48C3
ØØØØ4:72Ø2 4EAE FF28 613A 43FA Ø5CC 4EAE FE68:B7EE
ØØØØ5:23CØ ØØØØ Ø4C2 67ØØ ØØ28 2C4Ø 4EAE FFC4:ØA5C
ØØØØ6:6712 22ØØ 243C ØØØØ Ø212 263C ØØØØ Ø22F:D7CB
ØØØØ7:4EAE FFDØ 2C78 ØØØ4 227A Ø47C 4EAE FE62:EFØØ
ØØØØ8:4E75 41FA Ø241 72Ø7 E998 24ØØ Ø28Ø ØØØØ:14CF
ØØØØ9:ØØØF ØCØØ ØØØA 6DØ6 Ø6ØØ ØØ37 6ØØ4 Ø6ØØ:E55A
ØØØ1Ø:ØØ3Ø 1ØCØ 2ØØ2 51C9 FFEØ 4E75 ØC18 ØØ2Ø:DD48
ØØØ11:66Ø4 51C8 FFF8 43FA Ø56A 41E8 FFFF 534Ø:964F
ØØØ12:12D8 51C8 FFFC 2C78 ØØØ4 43FA Ø54A 4EAE:29ØA
ØØØ13:FE68 23CØ ØØØØ Ø4C2 67A6 2C4Ø 4EAE FFC4:Ø942
ØØØ14:23CØ ØØØØ Ø5DE 22ØØ 243C ØØØØ Ø441 263C:9A57
ØØØ15:ØØØØ ØØ3C 4EAE FFDØ Ø839 ØØØA ØØDF FØ16:47F2
ØØØ16:67ØØ Ø134 Ø839 ØØØ6 ØØBF EØØ1 66EA 2C79:E496
ØØØ17:ØØØØ Ø4C2 223C ØØØØ Ø5EE 243C FFFF FFFE:5125
ØØØ18:4EAE FFAC 23CØ ØØØØ Ø4C6 67ØØ Ø1ØA 2C79:ØB63
ØØØ19:ØØØØ Ø4C2 22ØØ 243C ØØØØ Ø4CA 4EAE FF9A:9E1Ø
ØØØ2Ø:4A8Ø 67ØØ ØØF2 41FA Ø3B6 2Ø28 ØØ7A 23CØ:3C84
ØØØ21:ØØØØ Ø5DA 223C ØØØ1 ØØØ2 2C78 ØØØ4 4EAE:A343
ØØØ22:FF3A 23CØ ØØØØ Ø5D6 67ØØ ØØCC 2C79 ØØØØ:BD15
ØØØ23:Ø4C2 223C ØØØØ Ø5EE 243C ØØØØ Ø3ED 4EAE:A3C3
ØØØ24:FFE2 23CØ ØØØØ Ø5CE 67ØØ ØØ98 22ØØ 2439:D741
ØØØ25:ØØØØ Ø5D6 2639 ØØØØ Ø5DA 4EAE FFD6 2239:A2A6
ØØØ26:ØØØØ Ø5CE 4EAE FFDC 2239 ØØØØ Ø5DE 243C:AØAB
ØØØ27:ØØØØ Ø47D 263C ØØØØ ØØ43 4EAE FFDØ Ø839:81B3
ØØØ28:ØØØA ØØDF FØ16 675A Ø839 ØØØ6 ØØBF EØØ1:4158
ØØØ29:66EC 223C ØØØØ Ø5EE 243C ØØØØ Ø3EE 4EAE:Ø5EE
ØØØ3Ø:FFE2 23CØ ØØØØ Ø5D2 67ØØ ØØ38 2239 ØØØØ:B2E5
ØØØ31:Ø5D2 2439 ØØØØ Ø5D6 2639 ØØØØ Ø5DA 4EAE:AAA2
ØØØ32:FFDØ 2239 ØØØØ Ø5D2 4EAE FFDC 2239 ØØØØ:989E
ØØØ33:Ø5DE 243C ØØØØ Ø4CØ 263C ØØØØ ØØØ2 4EAE:A3C6
ØØØ34:FFDØ 2279 ØØØØ Ø5D6 2Ø39 ØØØØ Ø5DA 2C78:7AAA
ØØØ35:ØØØ4 4EAE FF2E 2C78 ØØØ4 2279 ØØØØ Ø4C2:A197
ØØØ36:4EAE FE62 4E75 ØDØA 9B3Ø 3B33 313B 343Ø:E45D
ØØØ37:6D41 6E6F 7468 6572 2Ø7Ø 6F77 6572 6675:1158
ØØØ38:6C2Ø 746F 6F6C 2Ø64 6F6E 652Ø 6279 2Ø9B:C8Ø1
ØØØ39:343B 3333 3B34 3Ø6D 416C 646F 2Ø52 6573:FEAF
ØØØ4Ø:6574 9B3Ø 3B33 313B 343Ø 6D2F 9B31 3B33:E4D5
ØØØ41:323B 343Ø 6D4D 4354 9B3Ø 3B33 313B 343Ø:52DA
ØØØ42:6D2Ø 696E 2Ø31 3939 3Ø2E ØDØA ØA41 7661:EDD2
ØØØ43:696C 6162 6C65 2Ø43 6869 7Ø2D 4D65 6D6F:EAEØ
ØØØ44:7279 3A2Ø 24ØØ ØØØØ ØØØØ ØØØØ ØØ2Ø 6279:3332
ØØØ45:7465 732E ØDØA ØA48 6F77 2Ø74 6F2Ø 7573:7363
ØØØ46:652Ø 7468 652Ø 5343 4F5Ø 592Ø 636F 6D6D:ØB37
ØØØ47:616E 643A ØDØA 4A75 7374 2Ø67 6976 652Ø:7F98
ØØØ48:7468 652Ø 6669 6C65 6E61 6D65 2C2Ø 692E:1D6A
ØØØ49:652E 2Ø9B 373B 3331 3B34 3Ø6D 5343 4F5Ø:FE69
ØØØ5Ø:592Ø 6466 3Ø3A 632F 5365 746D 617Ø 9B3Ø:1561
ØØØ51:3B33 313B 343Ø 6DØD ØA48 6974 2Ø3C 5265:F4Ø8
```




```
00052:7475 726E 3E2C 7468 656E 2069 6E73 6572:F333
00053:7420 7468 6520 9B33 3B33 333B 3430 6D73:F8EC
00054:6F75 7263 6520 9B30 3B33 313B 3430 6D64:F02A
00055:6973 6B20 7768 6963 6820 6375 7272 656E:58D3
00056:746C 790D 0A63 6F6E 7461 696E 7320 7468:2CA1
00057:6520 6669 6C65 2E50 7265 7373 206D 6F75:DBF8
00058:7365 2D62 7574 746F 6E2C 2052 6967 6874:EB03
00059:2062 7574 746F 6E20 746F 2071 7569 742E:F6DC
00060:0D0A 596F 7572 2066 696C 6520 6973 206C:54BC
00061:6F61 6465 642C 7468 656E 2069 6E73 6572:0616
00062:7420 7468 6520 9B33 3B33 333B 3430 6D64:F8DD
00063:6573 7469 6E61 7469 6F6E 9B30 3B33 313B:33B2
00064:3430 6D20 6469 736B 2061 6E64 0D0A 646F:7962
00065:2061 7320 6265 666F 7265 2E20 596F 7572:CBBB
00066:2066 696C 6520 6973 206E 6F77 2063 6F70:781D
00067:6965 6420 6F6E 2074 6865 2064 6973 6B2E:BAD1
00068:0D0A 5761 726E 696E 673A 2064 6973 6B20:9C78
00069:6E61 6D65 2028 6578 2E20 6466 303A 2920:4D46
00070:6D75 7374 2062 6520 7370 6563 6966 6965:1209
00071:642E 0D0A 0A49 6E73 6572 7420 9B33 3B33:99EC
00072:313B 3430 6D73 6F75 7263 6520 9B30 3B33:F039
00073:313B 3430 6D64 6973 6B20 616E 6420 7072:DD62
00074:6573 7320 6D6F 7573 6520 6275 7474 6F6E:66EC
00075:2E0D 0A49 6E73 6572 7420 9B33 3B33 313B:87FC
00076:3430 6D64 6573 7469 6E61 7469 6F6E 209B:EE43
00077:303B 3331 3B34 306D 6469 736B 2061 6E64:35A6
00078:2070 7265 7373 206D 6F75 7365 2062 7574:9F65
00079:746F 6E2E 0D0A 0000 0000 0000 0000 0000:EFA7
00080:0000 0000 0000 0000 0000 0000 0000 0000:0000
00081:0000 0000 0000 0000 0000 0000 0000 0000:0000
00082:0000 0000 0000 0000 0000 0000 0000 0000:0000
00083:0000 0000 0000 0000 0000 0000 0000 0000:0000
00084:0000 0000 0000 0000 0000 0000 0000 0000:0000
00085:0000 0000 0000 0000 0000 0000 0000 0000:0000
00086:0000 0000 0000 0000 0000 0000 0000 0000:0000
00087:0000 0000 0000 0000 0000 0000 0000 0000:0000
00088:0000 0000 0000 0000 0000 0000 0000 0000:0000
00089:0000 0000 0000 0000 0000 0000 0000 0000:0000
00090:0000 0000 0000 0000 0000 0000 0000 0000:0000
00091:0000 0000 0000 0000 0000 0000 0000 0000:0000
00092:0000 0000 0000 0000 0000 0000 0000 0000:0000
00093:0000 0000 0000 0000 0000 0000 0000 0000:0000
00094:0000 0000 0000 0000 0000 0000 0000 0000:0000
00095:0000 0000 0000 0000 0000 0000 0000 0000:0000
00096:0000 0000 0000 0000 0000 0000 0000 0000:0000
00097:0000 0000 0000 646F 732E 6C69 6272 6172:07EA
00098:7900 0000 0000 0000 0000 0000 0000 0000:7900
00099:0000 0000 0000 0000 0000 0000 0000 0000:0000
00100:0000 0000 0000 0000 0000 0000 0000 0000:0000
00101:0000 0000 0000 0000 0000 0000 0000 0000:0000
00102:0000 0000 0000 0000 0000 0000 0000 0000:0000
00103:0000 7965 735F 636F 0000 03EC 0000 001F:543E
00104:0000 0000 0000 001E 0000 0032 0000 00A0:00F0
00105:0000 00AE 0000 00B6 0000 00DC 0000 00E2:0322
00106:0000 00F2 0000 00FC 0000 0104 0000 011C:040E
00107:0000 0130 0000 013A 0000 0140 0000 0150:04FA
00108:0000 015C 0000 0162 0000 016C 0000 0176:05A0
00109:0000 017C 0000 01A0 0000 01B0 0000 01BA:0686
00110:0000 01C0 0000 01C6 0000 01D0 0000 01DA:0730
00111:0000 01E0 0000 01F0 0000 01F6 0000 0208:07CE
00112:0000 0000 0000 03F2 0000 03EB 0000 0001:07DE
00113:0000 03F2 0000 0000 0006 9808 0000 7068:0C68
```

Du vrai cinéma

ANIMATIX

Créer des sprites multicolores c'est bien, les animer c'est mieux. Surtout par le biais d'un outil à la hauteur.

Cet utilitaire en GFA 3.0 permet la création et l'animation de sprites. Le second programme est un exemple d'utilisation des fichiers préalablement créés. L'éditeur d'Animatix propose l'élaboration de séquences d'une quinzaine de sprites (16 x 16, 24 x 24 ou 32 x 32 pixels). L'emploi est aisé par l'utilisation de menus déroulants dont voici le détail.

Project

• **Save:** sauvegarde la case en cours.

• **Load:** chargement d'une case.

• **Save animation:** sauvegarde d'une animation. Exemple,15 cases en une seule fois.

- Sélectionner «Animation select».
- Cliquer les 15 cases les unes après les autres.
- Sélectionner «Save animation».
- Entrer le nom du fichier («Essai» par exemple).
- Le programme sauvegardera alors les 15 cases sous les noms de «Essai1, Essai2,... Essais15).

• **Load animation:** charge une animation. Entrer le nom du fichier, plus le numéro 1. Pour recharger l'animation «Essai» de 1 à 15.

- Sélectionner «Load animation».
- Entrer «Essai1», le programme chargera le reste.

N. B. : pour «Save animation» et «Load animation», on doit impérativement avoir sélectionné au moins une animation, sinon, il ne se passera rien.

• **Palette:** lors de la sélection de cette option:

- Pour «Load animation», le programme chargera la palette de couleurs si elle existe.
- Pour «Save animation», le programme sauvegardera la palette de couleurs.

• **Quit:** quitter le programme.

Utile

• **Move:** déplace une case vers une autre. Après sélection, cliquer la case destination. La case destination est remplacée par la case actuelle qui s'efface.

• **Swap:** échange le contenu de 2 cases. Après sélection, cliquer la seconde case.

• **Copy:** duplique la case sur une autre. Après sélection, cliquer la case destination. Celle-ci est remplacée.

• **Inversion H:** inverse horizontalement le contenu de la case.

• **Inversion V:** inverse verticalement le contenu de la case.

• **Clear:** détruit la case actuelle.

Animation

• **Animation select:** sélection d'une animation. Sélectionner les cases les une après les autres (le programme n'accepte que 20 animations maxi). Quitter cette fonction en cliquant sur le bouton droit de la souris. Chaque nouvelle animation efface la précédente.

• **Animation start:** active l'animation sur l'écran. Si aucune n'a été sélectionnée, il ne se passera rien. Quitter cette fonction en cliquant longuement sur le bouton gauche. Régler la vitesse d'animation en déplaçant la souris de gauche à droite.

Format

• **16 x 16:** format des cases par défaut, correspondant à 16 pixels sur 16.

• **24 x 24:** correspond à 24 pixels/24.

• **32 x 32:** correspond à 32 pixels/32. Le plus grand format du programme.

• **Restore palette:** restaure les couleurs à leur état par défaut.

Divers

• Pour dessiner dans différentes cases, sélectionner la case en question.

• Pour chaque couleur sélectionnée, on peut modifier les niveaux de rouge, vert et bleu en cliquant dans les 3 cases réservées à cet effet.

• Le programme utilise les couleurs de 0 à 3. Ne pas les modifier afin d'éviter tout compromis avec l'affichage de présentation.

• Une petite case au-dessus de la palettisation indique:

- L'attente (Wait).
- Que l'on doit cliquer une autre case (Pick).
- Le numéro de sélection d'animation.

• Les cases sont sauvegardées sous le format «Get» et «Put» avec, en début de fichier, différents paramètres (voir commentaires du second programme).

• Il est possible de charger sur un format supérieur, des cases sauvegardées sous un format inférieur, mais pas l'inverse.

• Les images sont sauvegardées sous 5 plans de bits 32 couleurs.

Daniel Provenier

```
' ▪ indique l'endroit où
' vous devez frapper Return.
'
' ******************************▪
' *                            *▪
' *       ANIMATIX             *▪
' *                            *▪
' *       Writing by           *▪
' *                            *▪
' *       PROVENIER DANIEL     *▪
' *                            *▪
' *       LE  12/Ø2/199Ø       *▪
' ******************************▪
RESERVE 3ØØØØ▪
DEFWRD "a,z"▪
@init▪
EDIT▪
PROCEDURE principal▪
 FOR q=Ø TO 2▪
   @aff_palette(q)▪
 NEXT q▪
 ON MENU GOSUB quel_menu▪
 REPEAT▪
   ON MENU Ø.1▪
   MOUSE x,y,k▪
   IF k=1▪
     @clique▪
   ENDIF▪
 UNTIL fini=1▪
RETURN▪
PROCEDURE init▪
 DIM men$(25),co(31,2),an$(2Ø)▪
 di=1▪
 OPENS 1,Ø,Ø,32Ø,256,5,Ø▪
 OPENW 1,Ø,Ø,32Ø,256,Ø,Ø,1▪
 CLEARW 1▪
 TITLEW #1,""▪
```

```
 FOR n=Ø TO 25▪
   READ men$(n)▪
 NEXT n▪
 MENU men$()▪
 @restore_co▪
 @init2▪
 @get▪
 cl$=im$▪
 @principal▪
 pal=Ø▪
 valm%=&H52▪
 DATA Project,Save,Load,Save animation,L
oad animation,Palette,Quit,""▪
 DATA Utile,Move,Swap,Copy,Inversion.H,I
nversion.V,Clear,""▪
 DATA Anime,Animation select,Animation s
tart,""▪
 DATA Format,16*16,24*24,32*32,Restore p
alette,""▪
 palette:▪
 DATA Ø,Ø,Ø▪
 DATA 7,7,7▪
 DATA 5,15,15▪
 DATA 15,5,5▪
 DATA 15,15,15▪
 DATA 11,11,11▪
 DATA 6,6,6▪
 DATA 2,2,2▪
 DATA 15,Ø,Ø▪
 DATA 15,4,Ø▪
 DATA 15,8,Ø▪
 DATA 14,12,Ø▪
 DATA 13,14,Ø▪
 DATA 8,14,Ø▪
 DATA 4,14,Ø▪
 DATA Ø,14,Ø▪
 DATA Ø,15,Ø▪
 DATA Ø,15,4▪
 DATA Ø,15,8▪
 DATA Ø,14,12▪
 DATA Ø,12,14▪
 DATA Ø,8,14▪
 DATA Ø,4,14▪
 DATA Ø,Ø,14▪
 DATA Ø,Ø,15▪
 DATA 4,Ø,15▪
 DATA 9,Ø,15▪
 DATA 13,Ø,15▪
 DATA 15,Ø,13▪
 DATA 15,Ø,9▪
 DATA 15,Ø,4▪
 DATA 15,7,15▪
RETURN▪
PROCEDURE init1▪
 ALERT Ø,"ATTENTION|toutes les cases von
t etre|DETRUITES",1,"CONTINUE|OUBLIE !!"
,a▪
 IF a=1▪
   di=di1▪
   @init2▪
 ENDIF▪
RETURN▪
PROCEDURE init2▪
 DEFFILL 1,1,1▪
 PBOX Ø,Ø,32Ø,256▪
 @box(5,Ø,1Ø,192,2Ø2)▪
 @box(3,2ØØ,213,247,221)▪
 @box(3,2ØØ,228,247,236)▪
 @box(3,2ØØ,243,247,251)▪
 @box(3,Ø,218,194,244)▪
 @box(3,275,213,31Ø,223)▪
 COLOR 3▪
 FOR n=Ø TO 192 STEP 12+(4*(di=2))+(6*(d
i=3))▪
   LINE n,1Ø,n,2Ø2▪
   LINE Ø,n+1Ø,192,n+1Ø▪
 NEXT n▪
 COLOR 3▪
 BOX Ø,1Ø,192,2Ø2▪
 d1=(di-1)*8▪
 FOR n=Ø TO 2▪
   FOR m=Ø TO 4▪
     @box(3,2ØØ+(n*4Ø),1Ø+(m*4Ø),216+(di
-1)*8+(n*4Ø)+1,26+(m*4Ø)+(di-1)*8+1)▪
   NEXT m▪
 NEXT n▪
 c=Ø▪
 FOR n=Ø TO 1▪
   FOR m=Ø TO 15▪
     DEFFILL c,1,1▪
     PBOX m*12+1,n*12+22Ø,m*12+11,n*12+2
3Ø▪
     c=c+1▪
   NEXT m▪
 NEXT n▪
 COLOR 1▪
 c1=219▪
 c2=231▪
 c=Ø▪
 BOX Ø,219,12,231▪
 COLOR 2,1▪
 TEXT 253,221,"R"▪
 TEXT 253,236,"V"▪
 TEXT 253,251,"B"▪
 bt1=Ø▪
 bt2=Ø▪
 COLOR 3▪
 BOX 2ØØ,1Ø,216+(di-1)*8+1,26+(di-1)*8+1
▪
 nor=12+(4*(di=2))+(6*(di=3))▪
 n2=8+(8*di)▪
RETURN▪
PROCEDURE box(noir,x1,y1,x2,y2)▪
 DEFFILL Ø,1,1▪
 PBOX x1+noir,y1+noir,x2+noir,y2+noir▪
 DEFFILL Ø,1,1▪
 PBOX x1,y1,x2,y2▪
 DEFFILL 2▪
 BOX x1,y1,x2,y2▪
RETURN▪
PROCEDURE clique▪
 IF x<192 AND y<2Ø2 AND y>11▪
   @point▪
 ELSE IF x<194 AND y>218 AND y<244▪
   @choix_couleur▪
 ELSE IF x>2ØØ AND y>1Ø AND y<2ØØ▪
   @quel_boite▪
   @restit▪
 ELSE IF x>2ØØ AND x<247 AND y>213 AND y
<251▪
   @choix_palette▪
 ENDIF▪
RETURN▪
PROCEDURE point▪
```

```
 COLOR c▪
 x1=INT(x/nor)*nor▪
 y1=INT((y-1Ø)/nor)*nor+1Ø▪
 PBOX x1+1,y1+1,x1+nor-1,y1+nor-1▪
 PLOT 2Ø1+x1/nor+4Ø*bt1,11+INT(y1-1Ø)/no
r+4Ø*bt2▪
RETURN▪
PROCEDURE choix_couleur▪
 COLOR Ø▪
 BOX (c+(16*(c>15)))*12,c1,(c+(16*(c>15)
))*12+12,c2▪
 c=POINT(x,y)▪
 IF c<16▪
   c1=219▪
   c2=231▪
 ELSE▪
   c1=231▪
   c2=243▪
 ENDIF▪
 COLOR 1▪
 BOX (c+(16*(c>15)))*12,c1,(c+(16*(c>15)
))*12+12,c2▪
 FOR q=Ø TO 2▪
   @aff_palette(q)▪
 NEXT q▪
RETURN▪
PROCEDURE quel_boite▪
 IF x<24Ø▪
   bts1=Ø▪
 ELSE IF x<28Ø▪
   bts1=1▪
 ELSE▪
   bts1=2▪
 ENDIF▪
 IF y<5Ø▪
   bts2=Ø▪
 ELSE IF y<9Ø▪
   bts2=1▪
 ELSE IF y<13Ø▪
   bts2=2▪
 ELSE IF y<17Ø▪
   bts2=3▪
 ELSE▪
   bts2=4▪
 ENDIF▪
 COLOR 2▪
 BOX 2ØØ+(bt1*4Ø),1Ø+(bt2*4Ø),216+(di-1)
*8+(bt1*4Ø)+1,26+(bt2*4Ø)+(di-1)*8+1▪
 COLOR 3▪
 BOX 2ØØ+(bts1*4Ø),1Ø+(bts2*4Ø),216+(di-
1)*8+(bts1*4Ø)+1,26+(bts2*4Ø)+(di-1)*8+1
▪
 bt1=bts1▪
 bt2=bts2▪
RETURN▪
PROCEDURE restit▪
 COLOR 3,Ø▪
 TEXT 276,221,"WAIT"▪
 FOR n=Ø TO n2-1▪
   n1=n+2Ø1+(4Ø*bt1)▪
   n3=n*nor▪
   FOR m=Ø TO n2-1▪
     cp=POINT(n1,m+11+(4Ø*bt2))▪
     COLOR cp▪
     PBOX n3+1,(m*nor)+11,(n3+nor)-1,(m*
nor+nor)+9▪
   NEXT m▪
 NEXT n▪
 TEXT 276,221,"     "▪
RETURN▪
PROCEDURE quel_menu▪
 ON MENU(Ø) GOSUB save,load,save_an,load
_an,pale,quit,rien,rien,move,swap,copy,i
nverh,inverv,clear,rien,rien,ansel,ansta
rt,rien,rien,dim1,dim2,dim3,restore_co▪
RETURN▪
PROCEDURE quit▪
 fini=1▪
RETURN▪
PROCEDURE move▪
 COLOR 3,Ø▪
 TEXT 276,221,"PICK"▪
 @get▪
 c1$=im$▪
 im$=c1$▪
 test=Ø▪
 k=Ø▪
 REPEAT▪
   WHILE k<>1▪
     MOUSE x,y,k▪
   WEND▪
   IF x>2ØØ AND y>1Ø AND y<2ØØ▪
     @put▪
     im$=c1$▪
     @quel_boite▪
     test=1▪
   ENDIF▪
 UNTIL test=1▪
 @put▪
 @relache▪
 COLOR 3,Ø▪
 TEXT 276,221,"     "▪
RETURN▪
PROCEDURE swap▪
 COLOR 3,Ø▪
 TEXT 276,221,"PICK"▪
 @get▪
 c1$=im$▪
 test=Ø▪
 k=Ø▪
 REPEAT▪
   WHILE k<>1▪
     MOUSE x,y,k▪
   WEND▪
   IF x>2ØØ AND y>1Ø AND y<2ØØ▪
     bt11=bt1▪
     bt22=bt2▪
     @quel_boite▪
     @get▪
     SWAP bt1,bt11▪
     SWAP bt2,bt22▪
     @put▪
     SWAP bt1,bt11▪
     SWAP bt2,bt22▪
     im$=c1$▪
     @put▪
     test=1▪
   ENDIF▪
 UNTIL test=1▪
 @relache▪
 COLOR 3,Ø▪
 TEXT 276,221,"     "▪
RETURN▪
PROCEDURE copy▪
```

```
 COLOR 3,Ø▪
 TEXT 276,221,"PICK"▪
 @get▪
 test=Ø▪
 k=Ø▪
 REPEAT▪
   WHILE k<>1▪
     MOUSE x,y,k▪
   WEND▪
   IF x>2ØØ AND y>1Ø AND y<2ØØ▪
     @quel_boite▪
     @put▪
     test=1▪
   ENDIF▪
 UNTIL test=1▪
 @relache▪
 COLOR 3,Ø▪
 TEXT 276,221,"     "▪
RETURN▪
PROCEDURE clear▪
 ALERT Ø,"ATTENTION|cette case va etre|d
etruite",1,"CONTINUE|OUBLIE !!",a▪
 IF a=1▪
   im$=cl$▪
   @put▪
   @restit▪
 ENDIF▪
RETURN▪
PROCEDURE inverh▪
 n3=4Ø*bt2+11▪
 FOR n=Ø TO n2-1▪
   n1=n+2Ø1+(4Ø*bt1)▪
   FOR m=Ø TO (n2-1)/2▪
     cp1=POINT(n1,m+n3)▪
     cp2=POINT(n1,n3+n2-m-1)▪
     COLOR cp1▪
     PLOT n1,n3+n2-m-1▪
     COLOR cp2▪
     PLOT n1,m+n3▪
   NEXT m▪
 NEXT n▪
 @restit▪
 COLOR c▪
RETURN▪
PROCEDURE inverv▪
 n3=4Ø*bt2+11▪
 FOR n=Ø TO (n2-1)/2▪
   n1=n+2Ø1+(4Ø*bt1)▪
   FOR m=Ø TO n2-1▪
     cp1=POINT(n2+2ØØ+(4Ø*bt1)-n,m+n3)▪
     cp2=POINT(n1,m+n3)▪
     COLOR cp1▪
     PLOT n1,m+n3▪
     COLOR cp2▪
     PLOT n2+2ØØ+(4Ø*bt1)-n,m+n3▪
   NEXT m▪
 NEXT n▪
 @restit▪
 COLOR c▪
RETURN▪
PROCEDURE ansel▪
 ja=Ø▪
 btl1=bt1▪
 btl2=bt2▪
 WHILE MOUSEK<>2▪
   MOUSE x,y,k▪
   IF k=1▪
     IF x>2ØØ AND y>1Ø AND y<2ØØ AND ja<
2Ø▪
       @quel_boite▪
       @get▪
       an$(ja)=im$▪
       INC ja▪
       @relache▪
       COLOR 3,Ø▪
       TEXT 285,221,STR$(ja)▪
     ENDIF▪
   ENDIF▪
 WEND▪
 TEXT 285,221,"  "▪
 COLOR 2▪
 BOX 2ØØ+(bt1*4Ø),1Ø+(bt2*4Ø),216+(di-1)
*8+(bt1*4Ø)+1,26+(bt2*4Ø)+(di-1)*8+1▪
 COLOR 3▪
 bt1=btl1▪
 bt2=btl2▪
 BOX 2ØØ+(bt1*4Ø),1Ø+(bt2*4Ø),216+(di-1)
*8+(bt1*4Ø)+1,26+(bt2*4Ø)+(di-1)*8+1▪
RETURN▪
PROCEDURE anstart▪
 IF ja<>Ø▪
   GET 76,76,116,116,a$▪
   @box(3,76,76,1Ø9,1Ø9)▪
   WHILE MOUSEK<>1▪
     FOR n=Ø TO ja-1▪
       FOR tt=Ø TO MOUSEX*1Ø▪
       NEXT tt▪
       PUT 77,77,an$(n)▪
     NEXT n▪
   WEND▪
   PUT 76,76,a$▪
 ENDIF▪
RETURN▪
PROCEDURE get▪
 GET 2Ø1+(bt1*4Ø),11+(bt2*4Ø),216+(di-1)
*8+(bt1*4Ø),26+(bt2*4Ø)+(di-1)*8,im$▪
RETURN▪
PROCEDURE put▪
 PUT 2Ø1+(bt1*4Ø),11+(bt2*4Ø),im$▪
RETURN▪
PROCEDURE relache▪
 WHILE MOUSEK▪
 WEND▪
 FOR j=Ø TO 5ØØ▪
 NEXT j▪
 WHILE MOUSEK▪
 WEND▪
RETURN▪
PROCEDURE dim1▪
 IF di<>1▪
   di1=1▪
   @init1▪
 ENDIF▪
RETURN▪
PROCEDURE dim2▪
 IF di<>2▪
   di1=2▪
   @init1▪
 ENDIF▪
RETURN▪
PROCEDURE dim3▪
 IF di<>3▪
```

```
    di1=3▪
    @init1▪
  ENDIF▪
RETURN▪
PROCEDURE restore_co▪
  RESTORE palette▪
  FOR n=Ø TO 31▪
    FOR m=Ø TO 2▪
      READ co(n,m)▪
    NEXT m▪
    SETCOLOR n,co(n,Ø),co(n,1),co(n,2)▪
  NEXT n▪
  FOR q=Ø TO 2▪
    @aff_palette(q)▪
  NEXT q▪
RETURN▪
PROCEDURE aff_palette(q)▪
  SELECT q▪
  CASE Ø▪
    @box(3,2ØØ,213,247,221)▪
  CASE 1▪
    @box(3,2ØØ,228,247,236)▪
  CASE 2▪
    @box(3,2ØØ,243,247,251)▪
  ENDSELECT▪
  COLOR 3▪
  IF co(c,q)▪
    PBOX 2Ø1,214+15*q,2Ø1+3*co(c,q),214+1
5*q+6▪
  ENDIF▪
  COLOR c▪
RETURN▪
PROCEDURE choix_palette▪
  x1=FIX((x-2Ø1)/3)▪
  y1=y-214▪
  IF y1>=Ø AND y1<8▪
    IF co(c,Ø)<>x1▪
      co(c,Ø)=x1▪
      @aff_palette(Ø)▪
    ENDIF▪
  ELSE IF y1>=15 AND y1<23▪
    IF co(c,1)<>x1▪
      co(c,1)=x1▪
      @aff_palette(1)▪
    ENDIF▪
  ELSE IF y1>=3Ø AND y1<38▪
    IF co(c,2)<>x1▪
      co(c,2)=x1▪
      @aff_palette(2)▪
    ENDIF▪
  ENDIF▪
  SETCOLOR c,co(c,Ø),co(c,1),co(c,2)▪
RETURN▪
PROCEDURE save▪
  @get▪
  FILESELECT "SAUVE CASE","SAUVER","DFØ:"
,nom$▪
  IF nom$<>""▪
    a=LEN(im$)▪
    OPEN "O",#1,nom$▪
    PRINT #1,"ANIMATE"▪
    PRINT #1,a▪
    PRINT #1,di▪
    FOR n=1 TO a▪
      b=ASC(MID$(im$,n,1))▪
      OUT #1,b▪
    NEXT n▪
    CLOSE #1▪
  ENDIF▪
RETURN▪
PROCEDURE load▪
  FILESELECT "LOAD CASE","CHARGER","DFØ:"
,nom$▪
  IF nom$<>""▪
    OPEN "I",#1,nom$▪
    INPUT #1,sor$▪
    IF sor$="ANIMATE"▪
      INPUT #1,a▪
      INPUT #1,di1▪
      IF di1=<di▪
        im$=""▪
        FOR n=1 TO a▪
          b=INP(#1)▪
          im$=im$+CHR$(b)▪
        NEXT n▪
      ELSE▪
        ALERT Ø,"CETTE CASE|A ETE SAUVEGA
RDEE|SOUS UN FORMAT|SUPERIEUR",1," OK ",
a▪
      ENDIF▪
    ENDIF▪
    CLOSE▪
    @put▪
  ENDIF▪
RETURN▪
PROCEDURE load_an▪
  FILESELECT "LOAD ANIMATION","CHARGER","
DFØ:",nom$▪
  IF nom$<>"" AND RIGHT$(nom$,1)="1"▪
    nom$=MID$(nom$,1,LEN(nom$)-1)▪
    bts1=bt1▪
    bts2=bt2▪
    FOR bt1=Ø TO 2▪
      FOR bt2=Ø TO 4▪
        IF EXIST(nom$+STR$(bt2+(5*bt1)+1)
)▪
          OPEN "I",#1,nom$+STR$(bt2+(5*bt
1)+1)▪
          INPUT #1,sor$▪
          IF sor$="ANIMATE"▪
            INPUT #1,a▪
            INPUT #1,di1▪
            IF di1=<di▪
              im$=""▪
              FOR n=1 TO a▪
                b=INP(#1)▪
                im$=im$+CHR$(b)▪
              NEXT n▪
            ELSE▪
              ALERT Ø,"CETTE CASE|A ETE S
AUVEGARDEE|SOUS UN FORMAT|SUPERIEUR",1,"
 OK ",a▪
            ENDIF▪
          ENDIF▪
          CLOSE▪
          @put▪
        ENDIF▪
      NEXT bt2▪
    NEXT bt1▪
    IF pal=1▪
      IF EXIST(nom$+".PALETTE")▪
        OPEN "I",#1,nom$+".PALETTE"▪
```

```
      FOR n=Ø TO 31▪
        INPUT #1,co(n,Ø);co(n,1);co(n,2
)▪
        SETCOLOR n,co(n,Ø),co(n,1),co(n
,2)▪
      NEXT n▪
      CLOSE #1▪
    ENDIF▪
  ENDIF▪
  bt1=bts1▪
  bt2=bts2▪
  @restit▪
 ENDIF▪
RETURN▪
PROCEDURE save_an▪
 IF ja<>Ø▪
  FILESELECT "SAVE ANIMATION","SAUVER",
"DFØ:",nom$▪
  IF nom$<>""▪
    IF RIGHT$(nom$,1)="1"▪
      nom$=MID$(nom$,1,LEN(nom$)-1)▪
    ENDIF▪
    FOR m=Ø TO ja-1▪
      a=LEN(an$(m))▪
      OPEN "O",#1,nom$+STR$(m+1)▪
      PRINT #1,"ANIMATE"▪
      PRINT #1,a▪
      PRINT #1,di▪
      FOR n=1 TO a▪
        b=ASC(MID$(an$(m),n,1))▪
        OUT #1,b▪
      NEXT n▪
      CLOSE #1▪
    NEXT m▪
    IF pal=1▪
      OPEN "O",#1,nom$+".PALETTE"▪
      FOR n=Ø TO 31▪
        WRITE#1,co(n,Ø),co(n,1),co(n,2)
▪
      NEXT n▪
      CLOSE #1▪
    ENDIF▪
  ENDIF▪
 ENDIF▪
RETURN▪
PROCEDURE pale▪
 pal=pal XOR 1▪
 IF pal=1▪
  valm%=valm% OR &H151▪
 ELSE IF pal=Ø▪
  valm%=valm% AND &HFE▪
 ENDIF▪
 MENU (5),valm%▪
RETURN▪
```

```
' ▪ indique l'endroit où
' vous devez frapper Return.
'
RESERVE 1ØØØØ▪
DEFWRD "a,z"▪
@init▪
@load_an▪
@animation_start▪
CLOSES 1▪
EDIT▪
PROCEDURE init▪
 DIM an$(2Ø)▪
 OPENS 1,Ø,Ø,32Ø,256,5,Ø▪
 OPENW 1,Ø,Ø,32Ø,256,Ø,Ø,1▪
 CLEARW 1▪
 TITLEW #1,""▪
RETURN▪
PROCEDURE animation_start▪
 IF ja<>Ø▪
  WHILE MOUSEK=Ø▪
    FOR n=Ø TO ja-1▪
      FOR tt=Ø TO MOUSEX*5 !--->VARIATI
ON DE LA VITESSE AVEC LA SOURIS▪
      NEXT tt▪
      PUT 77,77,an$(n)▪
    NEXT n▪
  WEND▪
  CLEARW 1▪
 ENDIF▪
RETURN▪
PROCEDURE load_an▪
 FILESELECT "LOAD ANIMATION","CHARGER","
DFØ:",nom$▪
 IF nom$<>"" AND RIGHT$(nom$,1)="1" ! RE
NTRE LE NOM DE LA 1 er IMAGE▪
  nom$=MID$(nom$,1,LEN(nom$)-1)     ! LE
 PROGRAME CHARGERA LE RESTE▪
  FOR m=Ø TO 14▪
    IF EXIST(nom$+STR$(m+1))▪
      OPEN "I",#1,nom$+STR$(m+1)▪
      INPUT #1,sor$       !  -----> VERIF
ICATION SI BON FICHIER▪
      IF sor$="ANIMATE"   !  -----> LE PR
EMIER CODE EST "ANIMATE"▪
        INPUT #1,a        !  -----> LONGU
EUR DU FICHIER▪
        INPUT #1,di1      !  -----> TEST
POUR LE FORMAT D'IMAGE▪
        '                                1=
16*16▪
        '                                2=
24*24▪
        '                                3=
32*32▪
        FOR n=1 TO a      !  -----> CHARG
EMENT DE L'IMAGE▪
          b=INP(#1)▪
          an$(ja)=an$(ja)+CHR$(b)▪
        NEXT n▪
        ja=ja+1           !   --->NOMBRE
D'IMAGES▪
      ENDIF▪
      CLOSE #1▪
    ENDIF▪
  NEXT m▪
  IF EXIST(nom$+".PALETTE")  ! ---> VER
IFICATION POUR SAVOIR SI LA PALETTE▪
    OPEN "I",#1,nom$+".PALETTE" !     ES
T PRESENTE SUR LA DISQUETTE▪
    FOR n=Ø TO 31▪
      INPUT #1,c1,c2,c3▪
      SETCOLOR n,c1,c2,c3▪
    NEXT n▪
    CLOSE #1▪
  ENDIF▪
 ENDIF▪
RETURN▪
```

Plein gaz!

UNE MANETTE DES GAZ SUR PC

Dingues des simulateurs de vol et mordus de jeux vidéos, ce montage est fait pour vous. Modique (60 F environ) et facile à réaliser, il permet le contrôle aisé de l'engin que vous pilotez. Condition sine qua non: que le jeu prenne en compte ce périphérique supplémentaire.

Le principe du montage est simple: Une carte Joystick possède quatre ports. Deux sont utilisés dans un joystick classique (coord. X et Y), et donc ... deux sont inutilisés! Deux? non, un est réservé par certains jeux pour contrôler la vitesse. Lesquels? Ceux compatibles *Maxx*™ (un manche à balai superbe ... mais cher et qui possède un contrôle des gaz) soit, entre autres, *Flight Simulator III*™, *Chuck Yeager's advanced flight trainer*™, *Silent Service*™, etc.

La Carte Joystick

Pour brancher une manette de jeux, il faut utiliser une carte adaptateur, laquelle autorise quatre entrées analogiques (c'est-à-dire peuvant prendre des valeurs de 0 à n) et deux entrées de type «tout ou rien». En théorie, cette carte est prévue pour deux joysticks, chacun devant donc utiliser deux entrées analogiques, ainsi qu'un entrée numérique. En fait, un joystick utilise deux boutons de mise à feu et occupe de ce fait les deux entrées «tout ou rien». Le principe de mesure des entrées analogiques est le suivant: aux bornes de chacune d'elles se trouve un potentiomètre (résistance variable). Suivant la position de ce dernier, la résistance varie et est évaluée par une temporisation plus ou moins longue mesurée à tout instant par le jeu.

Principe

Il suffit d'intercaler un potentiomètre entre les bornes de la troisième entrée analogique et le tour est joué! Mais voyons d'abord les différentes cartes Joystick. En fait, deux types de cartes: les cartes un port et les cartes deux ports qui permettent le branchement d'un second joystick.

Il est beaucoup plus facile d'exécuter le montage sur un carte deux ports. En effet, le premier étant pris par le joystick, le montage se fera sur le deuxième, ce qui est plus simple à réaliser.

Si l'on regarde le premier schéma (Carte joystick 1 port), on peut noter qu'il y a une rangée de broches réservée pour le joystick 1 et une autre pour le joystick 2. Oui, mais si l'on peut brancher un joystick (semblable aux autres) sur le second port, c'est que le brochage est inversé (d'où le second schéma)

Réalisation

Sans danger pour l'ordinateur, elle ne pose aucun problème pour quiconque n'est pas trop maladroit. Toutefois, il est conseillé d'utiliser un fer à souder de faible puissance (inférieure à 50 W) et de la soudure électronique à 60%.

D'abord, il faut souder les deux brins (fils) du câble sur les broches 1 et 6 de la prise db15. Inutile de compter, les broches sont numérotées. Si possible, utiliser un étau pour fixer le connecteur. Sinon, faites tenir la prise par un copain obligeant armé d'une paire de pinces isolantes (gaffe aux brûlures!).

A l'autre extrémité du câble «scotchez» les brins sur deux pattes côte à côte du potentiomètre. Branchez et essayez (sur un jeu qui accepte). Ca marche? Alors soudez le potentiomètre. Dans le cas contraire, vérifiez le montageet assurez-vous que la manette est au zéro (Sinon, *Chuck Yeager*, par exemple, ne la reconnaîtra pas). De toute façon, ÇA DOIT MARCHER!!! Il ne reste plus qu'à loger votre manette dans un boîtier (ou la fixer au clavier), puis jouer, enfin.

Facile non? Hélas, trop peu de jeux exploitent cette possibilité. Imaginez une bourre sur *Falcon*™ avec ce genre d'engin... A vous les manœuvres Top Gun ... Quand *Spectrum Holobyte* daignera le prévoir! De plus, il reste le quatrième port. Il n'existe pas de norme établie, mais *Flight Simulator* (3 et 4) utilise ledit port pour le palonnier (les pédales qui doivent être conjuguées avec le manche).... Fanas de réalisme et bidouilleurs de tous poils, vous savez ce qu'il vous reste à faire.

Jean-François Six

Liste du matériel

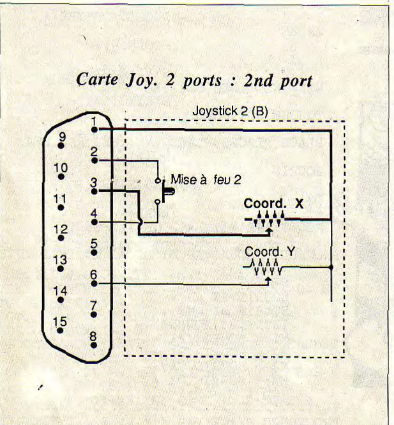
- 1m 50 de câble deux brins.
- 1 potentiomètre droit linéaire variant de 0 à 100 kilo-ohms (kΩ).
- 1 connecteur mâle DB 15 (15 broches).

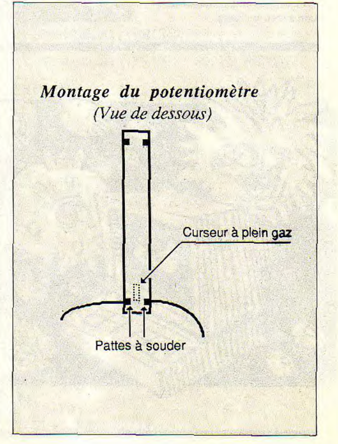
Accessoirement :

- 1 capot pour DB15
- 1 boîtier pour le potentiomètre.

Carte joystick 1 port

Joystick 2 (B)
Joystick 1 (A)
Mise à feu 2
Coord. X
Coord. Y
Mise à feu 1
Coord. X
Coord. Y
1 2 3 4 5 6 7 8
9 10 11 12 13 14 15





Sus aux bagnoles!

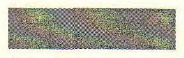
MAD DAV

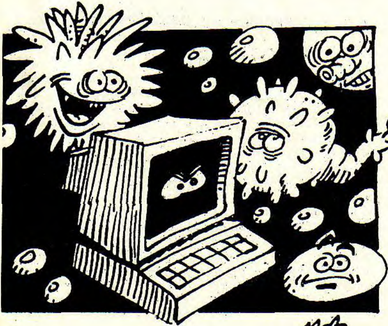
Excellent exutoire! Rouler à fond la caisse aux commandes d'un bolide, en canardant tout ce qui roule...

Programmé en Turbo Pascal (4 ou 5), ce jeu, tournant en mode graphique CGA minimum, nécessite dans le même répertoire les fichiers CGA.BGI, GOTH.CHR, LITT.CHR et GRAPH.TPU.
Il se joue au joystick (ne choisir cette option qu'en possession de la carte sous peine de plantage), à la souris ou avec les chiffres. Les indications nécessaires sont incluses.

On ne peut heurter plus de 10 fois le bord de la route. Les forces de 1 à 4 offrent respectivement un bonus de 10 à 25 points. La touche ESC, comme à son habitude, permet de sortir.

David Farenzena





```
' ▪ indique l'endroit où
' vous devez frapper Return.
'
{******** MAD DAV *********}▪
{*******EN TP4 OU 5********}▪
{*****FARENZENA DAVID******}▪
PROGRAM MAD_DAV;▪
USES CRT,GRAPH,DOS;▪
VAR▪
  X,Y,XX,YY,X1,Y1,XX2,YY2,XX3,XX4,YY3,YY
4,U,V,A,B,B1,B2       :INTEGER;▪
  I,I2,J,K,L,M,J1,SC,SC1,KL,KL1,KL2,KL3,
SCT,SCO,SCOR          :INTEGER;▪
  O,O1,O2,O4,O5,P,Q,R,S,T,T1,LEN,DRI,MOD
E,COMPT,MUST          :INTEGER;▪
  TAILLE,TAILLE2,TAILLE3,TV1,TV2,TV3,TV4
,FORCE                :WORD;▪
  HEU,MIN,SEC,CEN,JOY1,BO,SC2,SC3
                      :WORD;▪
  CO,COI
                      :ARCCOORDSTYPE;▪
  ZA,ZE
                      :BOOLEAN;▪
  L_,T_,V_,R_
                      :STRING[3];▪
  MIN_,SEC_,NOM,SC1_,KL
                      :STRING[10];▪
  TOUCHE,JOY,FORCE_
                      :CHAR;▪
  PLACE,PLACE2,PLACE3,PF1,PF2,PF3,PF4
                      :POINTER;▪
  SOURIS
                      :REGISTERS;▪
  O3
                      :REAL;▪
  SCORT:ARRAY[1..40] OF INTEGER;▪
{--------------------------------}▪
PROCEDURE SOUR(VAR M1,M2,M3,M4:INTEGER);
▪
      BEGIN▪
      SOURIS.AX:=3;▪
      SOURIS.BX:=M2;▪
      INTR($33,SOURIS);▪
      M1:= SOURIS.AX;▪
      M2:= SOURIS.BX;▪
      M3:= SOURIS.CX;▪
      M4:= SOURIS.DX;▪
      END;▪
{--------------------------------}▪
PROCEDURE STICK(VAR X,Y,K1,K2: INTEGER);
▪
```

```
     BEGIN▪
     INLINE▪
     ($BA/$Ø1/$Ø2/$EC/$A8/$Ø3/$75/$FB/$B
9/$2C/$Ø1/$FA/$EE/$EC/$A8/$Ø1/▪
     $EØ/$FB/$BB/$2C/$Ø1/$2B/$D9/$C4/$BE
/X/$26/$89/$1D/$2B/$DB/$A8/$1Ø/▪
     $75/$Ø3/$83/$C3/$Ø1/$C4/$BE/K1/$26/
$89/$1D/$2B/$DB/$A8/$2Ø/$75/$Ø3/▪
     $83/$C3/$Ø1/$C4/$BE/K2/$26/$89/$1D/
$EC/$A8/$Ø3/$75/$FB/$B9/$2C/$Ø1/▪
     $EE/$EC/$A8/$Ø2/$EØ/$FB/$FB/$BB/$2C
/$Ø1/$2B/$D9/$C4/$BE/Y/▪
     $26/$89/$1D);▪
     END;▪
{---------------------------------}▪
  PROCEDURE CADRE;▪
VAR I:INTEGER;▪
  BEGIN▪
  GOTOXY(1,1);WRITE(CHR(2Ø1));▪
  FOR I:=2 TO 79 DO▪
  BEGIN▪
  GOTOXY(I,1);WRITE(CHR(2Ø5));▪
  END;▪
  GOTOXY(8Ø,1);WRITE(CHR(187));▪
  FOR I:=2 TO 23 DO▪
  BEGIN▪
  GOTOXY(8Ø,I);WRITE(CHR(186));▪
  END;▪
  GOTOXY(8Ø,24);WRITE(CHR(188));▪
  FOR I:=2 TO 79 DO▪
  BEGIN▪
  GOTOXY(I,24);WRITE(CHR(2Ø5));▪
  END;▪
  GOTOXY(1,24);WRITE(CHR(2ØØ));▪
  FOR I:=2 TO 23 DO▪
  BEGIN▪
  GOTOXY(1,I);WRITE(CHR(186));▪
  END;▪
  END;▪
{---------------------------------}▪
PROCEDURE ENTREE;▪
 BEGIN▪
 TEXTCOLOR(1);▪
 CADRE;▪
 TEXTCOLOR(4);▪
 GOTOXY(36,1);WRITE('MAD DAV');▪
 GOTOXY(35,2);WRITE('=========');▪
 TEXTCOLOR(2);▪
 GOTOXY(4,4);WRITE('VOUS ETES L''AIGLE D
E LA ROUTE DETERMINE A ROULER LE PLUS VI
TE');▪
 GOTOXY(4,5);WRITE('POSSIBLE ET A PARCOU
RIR LE PLUS GRAND NOMBRE DE KILOMETRES.'
);▪
 GOTOXY(4,6);WRITE('POUR CELA VOUS ETES
PRET A DETRUIRE TOUT ENNEMI VENANT EN SE
NS INVERSE.');▪
 GOTOXY(4,7);WRITE('ATTENTION : PAS PLUS
 DE DIX SORTIES DE ROUTE.');▪
 GOTOXY(4,8);WRITE('LES ENNEMIS VIENNENT
 DE PLUS EN PLUS VITE.');▪
 GOTOXY(4,9);WRITE('POUR LES TUER,VOUS D
EVEZ LES PULVERISER AU RADIANT.');▪
 GOTOXY(4,16);WRITE('(ESC)POUR SORTIR.')
;▪
 TEXTCOLOR(5);▪
 GOTOXY(4,11);WRITE('AVEC UN JOYSTICK OU
 A LA SOURIS :(UN DES BOUTONS) POUR TIRE
R.');▪
 GOTOXY(4,12);WRITE('(LA MANETTE OU MA S
OURIS VERS SOI) POUR FREINER.');▪
 GOTOXY(4,13);WRITE('AVEC LES CHIFFRES:(
7 OU 9) POUR TIRER.(8) POUR AVANCER.(2)
POUR FREINER.');▪
 GOTOXY(4,14);WRITE('(6) A DROITE.(4) A
GAUCHE.(5) RETOUR AU MILIEU.');▪
 TEXTCOLOR(3);▪
 GOTOXY(16,17);WRITE('VOTRE FORCE (1/4):
');▪
 READLN(FORCE_);▪
 CASE FORCE_ OF▪
 '1':FORCE:=25;▪
 '2':FORCE:=2Ø;▪
 '3':FORCE:=15;▪
 '4':FORCE:=1Ø;▪
 ELSE BEGIN FORCE_:='4';FORCE:=1Ø;END;▪
 END;▪
 GOTOXY(3Ø,17);WRITE('         ');▪
 GOTOXY(16,17);WRITE('VOTRE PRENOM      :
');▪
 READLN(NOM);▪
 GOTOXY(16,17);WRITE('
');▪
 LEN:=(LENGTH(NOM)) DIV 2;▪
 GOTOXY(3Ø-LEN,17);WRITE('CHER(E) ',NOM,
' AVEZ VOUS ');▪
 GOTOXY(3Ø-LEN,18);WRITE('UN JOYSTICK
 :(1)');▪
 GOTOXY(3Ø-LEN,19);WRITE('UNE SOURIS
 :(2)');▪
 GOTOXY(3Ø-LEN,2Ø);WRITE('LES CHIFFRES
 :(3)');▪
 TEXTCOLOR(4);▪
 GOTOXY(3Ø-LEN,21);WRITE('--------------
-----');▪
 GOTOXY(3Ø-LEN,22);WRITE('VOTRE REPONSE
 : ');▪
 READLN(JOY);▪
 CASE JOY OF▪
 '1':JOY1:=1;▪
 '2':JOY1:=2;▪
 '3':JOY1:=3;▪
 ELSE JOY1:=3;▪
 END;▪
 TEXTCOLOR(6);▪
 GOTOXY(29-LEN,24);WRITE(NOM,' FRAPPEZ
  (ENTER) ');▪
 READLN;▪
 END;▪
{---------------------------------}▪
PROCEDURE SORTIE;▪
VAR J:STRING;I,CRA:INTEGER;▪
 BEGIN▪
 INC(COMPT);▪
 CASE FORCE_ OF▪
 '1':CRA:=1Ø;▪
 '2':CRA:=15;▪
 '3':CRA:=2Ø;▪
 '4':CRA:=25;▪
 END;▪
 SCO:=(SC*1Ø);▪
 SCOR:=KL+(CRA-SCT);▪
```

```
SCORT[COMPT]:=SCOR;▪
{***SYSTEME DE TRI***}▪
FOR I:=1 TO COMPT DO▪
BEGIN▪
IF SCORT[I]>SCORT[COMPT] THEN  MUST:=SC
ORT[I];▪
END;▪
IF MUST<SCOR THEN MUST:=SCOR;▪
CASE JOY1 OF▪
1:J:='UN JOYSTICK';▪
2:J:='UNE SOURIS';▪
3:J:='LES FLECHES';▪
END;▪
CLRSCR;▪
CADRE;▪
GOTOXY(17-LEN,4);▪
WRITE('FORCE= ',FORCE_,' SOIT UN BONUS
DE ',CRA,' POINTS.');▪
GOTOXY(17-LEN,6);▪
WRITE('MEILLEUR SCORE=',MUST,' POINTS.'
);▪
GOTOXY(17-LEN,8);▪
WRITE('VOTRE    SCORE=',SCOR,' POINTS.'
);▪
TEXTCOLOR(2);▪
GOTOXY(36,1);WRITE('MAD DAV');▪
GOTOXY(35,2);WRITE('=========');▪
GOTOXY(17-LEN,1Ø);▪
WRITE('CHER(E) ',NOM,' VOUS AVEZ PARCOU
RU ',KL,' KILOMETRES ',SCO,' METRES');▪
GOTOXY(17-LEN,12);▪
WRITE('EN COGNANT ',SCT,' FOIS LA ROUTE
,VOUS AVEZ TUE ',KL,' ENNEMIS.');▪
GOTOXY(17-LEN,14);▪
WRITE('ET TOUT CELA AVEC ',J,' SALUT AI
GLE DE LA ROUTE.');▪
TEXTCOLOR(4);▪
GOTOXY(3Ø-LEN,23);WRITE('UNE AUTRE ',NO
M,' (O/N) :');▪
READLN(TOUCHE);▪
END;▪
{--------------------------------}▪
PROCEDURE DESSIN_MOB;▪
BEGIN▪
{---DESSINS DES MOBILES---}▪
{---ENNEMI1---}▪
SETFILLSTYLE(1,2);▪
SETCOLOR(2);▪
RECTANGLE(158,3Ø,162,36);RECTANGLE(159,
31,161,33);▪
SETCOLOR(3);▪
RECTANGLE(155,33,157,37);RECTANGLE(163,
33,165,37);▪
SETCOLOR(1);PUTPIXEL(159,35,1);PUTPIXEL
(161,35,1);▪
PUTPIXEL(157,32,1);PUTPIXEL(163,32,1);▪
TV1:=IMAGESIZE(155,3Ø,165,37);▪
GETMEM(PF1,TV1);▪
GETIMAGE(155,3Ø,165,37,PF1^);▪
PUTIMAGE(155,3Ø,PF1^,1);▪
{---ENNEMI2---}▪
SETCOLOR(2);▪
RECTANGLE(154,4Ø,166,58);RECTANGLE(157,
43,163,49);▪
FLOODFILL(16Ø,53,2);▪
SETCOLOR(3);▪
RECTANGLE(145,52,153,61);RECTANGLE(167,
52,175,61);▪
SETCOLOR(1);CIRCLE(157,55,2);CIRCLE(163
,55,2);▪
FILLELLIPSE(151,46,2,2);FILLELLIPSE(169
,46,2,2);▪
TV2:=IMAGESIZE(145,4Ø,175,61);▪
GETMEM(PF2,TV2);▪
GETIMAGE(145,4Ø,175,61,PF2^);▪
PUTIMAGE(145,4Ø,PF2^,1);▪
{---ENNEMI3---}▪
SETCOLOR(2);▪
RECTANGLE(142,5Ø,178,1Ø4);RECTANGLE(151
,59,169,75);▪
FLOODFILL(16Ø,86,2);▪
SETCOLOR(3);▪
RECTANGLE(115,86,141,113);RECTANGLE(179
,86,2Ø5,113);▪
SETCOLOR(1);CIRCLE(151,95,6);CIRCLE(169
,95,6);▪
FILLELLIPSE(133,68,6,6);FILLELLIPSE(187
,68,6,6);▪
TV3:=IMAGESIZE(115,5Ø,2Ø5,113);▪
GETMEM(PF3,TV3);▪
GETIMAGE(115,5Ø,2Ø5,113,PF3^);▪
PUTIMAGE(115,5Ø,PF3^,1);▪
{---ENNEMI4---}▪
SETCOLOR(2);▪
SETLINESTYLE(Ø,Ø,THICKWIDTH);▪
RECTANGLE(13Ø,7Ø,19Ø,16Ø);RECTANGLE(145
,85,175,115);▪
FLOODFILL(16Ø,13Ø,2);▪
SETCOLOR(3);▪
RECTANGLE(85,13Ø,128,175);RECTANGLE(192
,13Ø,235,175);▪
SETCOLOR(1);CIRCLE(145,145,9);▪
SETLINESTYLE(Ø,Ø,THICKWIDTH);CIRCLE(175
,145,9);▪
FILLELLIPSE(115,1ØØ,9,9);FILLELLIPSE(2Ø
5,1ØØ,9,9);▪
TV4:=IMAGESIZE(84,69,236,176);▪
GETMEM(PF4,TV4);▪
GETIMAGE(84,69,236,176,PF4^);▪
PUTIMAGE(84,69,PF4^,1);▪
{---VOLANT DROIT---}▪
SETCOLOR(1);▪
CIRCLE(16Ø,195,55);▪
CIRCLE(16Ø,195,6Ø);▪
SETCOLOR(2);▪
SETLINESTYLE(Ø,Ø,THICKWIDTH);▪
LINE (16Ø,195,13Ø,16Ø);▪
FILLELLIPSE(16Ø,197,8,6);▪
TAILLE:=IMAGESIZE(1ØØ,14Ø,225,199);▪
GETMEM(PLACE,TAILLE);▪
GETIMAGE(1ØØ,14Ø,225,199,PLACE^);▪
PUTIMAGE(1ØØ,14Ø,PLACE^,1);▪
{---VOLANT GAUCHE---}▪
SETCOLOR(1);▪
CIRCLE(16Ø,195,55);▪
CIRCLE(16Ø,195,6Ø);▪
SETCOLOR(2);▪
SETLINESTYLE(Ø,Ø,THICKWIDTH);▪
LINE(16Ø,195,19Ø,16Ø);▪
FILLELLIPSE(16Ø,197,8,6);▪
TAILLE2:=IMAGESIZE(1ØØ,14Ø,225,199);▪
GETMEM(PLACE2,TAILLE2);▪
```

```
GETIMAGE(1ØØ,14Ø,225,199,PLACE2^);▪
PUTIMAGE(1ØØ,14Ø,PLACE2^,1);▪
{---VOLANT CENTRE---}▪
SETCOLOR(1);▪
CIRCLE(16Ø,195,55);▪
CIRCLE(16Ø,195,6Ø);▪
SETCOLOR(2);▪
SETLINESTYLE(Ø,Ø,THICKWIDTH);▪
LINE (16Ø,195,214,19Ø);▪
LINE (16Ø,195,1Ø6,19Ø);▪
FILLELLIPSE(16Ø,197,8,6);▪
TAILLE3:=IMAGESIZE(1ØØ,14Ø,225,199);▪
GETMEM(PLACE3,TAILLE3);▪
GETIMAGE(1ØØ,14Ø,225,199,PLACE3^);▪
PUTIMAGE(1ØØ,14Ø,PLACE3^,1);▪
END;▪
{---------------------------------}▪
PROCEDURE INITIALISATION;▪
BEGIN▪
X:=16Ø;Y:=17Ø;ZA:=TRUE;TOUCHE:=#Ø;I:=33
Ø;K:=12;L:=11;L_:=' ';U:=1;▪
A:=14;M:=Ø;B:=Ø;XX2:=14Ø;YY2:=1;O:=Ø;P:
=Ø;Q:=Ø;R:=1;T:=1;S:=1;▪
V:=Ø;DRI:=1;MODE:=2;J1:=Ø;X1:=18Ø;Y1:=1
;O1:=Ø;SC:=Ø;SC1:=1Ø;KL:=Ø;▪
SC1_:='1Ø';KL_:='Ø';BO:=1;SC2:=Ø;SC3:=Ø
;T1:=1;KL1:=Ø;KL2:=Ø;KL3:=Ø;▪
SCT:=Ø;O5:=Ø;SCO:=Ø;SCOR:=Ø;▪
END;▪
{---------------------------------}▪
PROCEDURE TACHI;▪
BEGIN▪
IF ZA AND (I>2Ø) THEN▪
BEGIN▪
DEC(I,5);▪
END;▪
IF (NOT ZA) AND (I<32Ø) THEN INC (I,5);
▪
SOUND (41Ø-I);DELAY(1Ø);NOSOUND;▪
SETCOLOR(Ø);▪
LINE(CO.XEND,CO.YEND,CO.X,CO.Y);▪
END;▪
{---------------------------------}▪
PROCEDURE DESSIN_ECRAN;▪
BEGIN▪
{--DESSIN DU TACHIMETRE--}▪
SETTEXTJUSTIFY(1,1);▪
SETTEXTSTYLE(2,Ø,2);▪
SETCOLOR(3);▪
FOR M:=6 TO 8 DO  CIRCLE (X,Y,A+M);▪
{---DESSIN VOITURE---}▪
SETCOLOR(2);▪
SETFILLSTYLE(7,3);▪
SETLINESTYLE(Ø,Ø,THICKWIDTH);▪
BAR(35,15Ø,8Ø,199);BAR(24Ø,15Ø,285,199)
;▪
RECTANGLE(35,15Ø,8Ø,2ØØ);RECTANGLE(24Ø,
15Ø,285,2ØØ);▪
MOVETO(9Ø,199);LINETO(9Ø,175);LINETO(12
Ø,15Ø);MOVETO(2ØØ,15Ø);▪
LINETO(23Ø,175);LINETO(23Ø,199);ELLIPSE
(16Ø,15Ø,Ø,18Ø,39,9);▪
SETCOLOR(1);▪
SETLINESTYLE(Ø,Ø,THICKWIDTH);▪
LINE(1,199,319,199);▪
SETFILLSTYLE(1,1);▪
SETLINESTYLE(Ø,Ø,Ø);▪
BAR(83,19Ø,87,195);BAR(233,19Ø,237,195)
;▪
SETCOLOR(3);▪
LINE(1,19,32Ø,19);▪
{---DESSIN DES COMPTEURS---}▪
SETCOLOR(2);▪
SETFILLSTYLE(1,1);▪
FILLELLIPSE(194,182,12,7);▪
OUTTEXTXY(194,182,SC1_);▪
OUTTEXTXY(194,17Ø,'CRACH');▪
FILLELLIPSE(126,182,12,7);▪
OUTTEXTXY(126,182,KL_);▪
OUTTEXTXY(126,17Ø,'KILO');▪
WHILE K<36Ø DO▪
BEGIN▪
SETCOLOR(Ø);▪
ARC(X,Y,K,K+1,A);▪
GETARCCOORDS(CO);▪
IF K>8Ø THEN SETCOLOR(1) ELSE SETCOLOR(
2);▪
INC(K,3Ø);▪
STR(L,L_);▪
OUTTEXTXY(CO.XEND,CO.YEND,L_);▪
DEC(L);▪
END;▪
ZE:=FALSE;▪
END;▪
{---PROCEDURES DE DESSIN DES VOLANTS---}
▪
PROCEDURE VOLANTC;▪
BEGIN▪
IF (B=3) THEN PUTIMAGE(1ØØ,14Ø,PLACE^,1
);▪
IF (B=2) THEN PUTIMAGE(1ØØ,14Ø,PLACE2^,
1);▪
PUTIMAGE(1ØØ,14Ø,PLACE3^,1);▪
B:=1;▪
END;▪
{---------------------------------}▪
PROCEDURE VOLANTD;▪
BEGIN▪
IF R<4 THEN INC(R) ELSE R:=1;▪
IF B=1 THEN PUTIMAGE(1ØØ,14Ø,PLACE3^,1)
;▪
PUTIMAGE(1ØØ,14Ø,PLACE^,1);▪
B:=3;▪
END;▪
{---------------------------------}▪
PROCEDURE VOLANTG;▪
BEGIN▪
IF R>1 THEN DEC(R) ELSE R:=4;▪
IF B=1 THEN PUTIMAGE(1ØØ,14Ø,PLACE3^,1)
;▪
PUTIMAGE(1ØØ,14Ø,PLACE2^,1);▪
B:=2;▪
END;▪
{---------------------------------}▪
PROCEDURE ROUTE1;▪
BEGIN▪
SETCOLOR(1);▪
SETLINESTYLE(3,Ø,Ø);▪
LINE(1,165,XX2+O,2Ø);▪
LINE(319,165,X1+O,2Ø);▪
END;▪
{---------------------------------}▪
```

```
PROCEDURE ROUTEØ;▪
 BEGIN▪
 SETCOLOR(Ø);▪
 LINE(1,165,XX2+O,2Ø);▪
 LINE(319,165,X1+O,2Ø);▪
 END;▪
{-----------------------------------}▪
PROCEDURE FLECHES;▪
 BEGIN▪
 CASE TOUCHE OF▪
 '8':BEGIN YY:=1Ø;XX:=16Ø;END;▪
 '2':BEGIN YY:=195;I:=32Ø;END;▪
 '4':XX:=5;▪
 '6':XX:=3ØØ;▪
 '5':XX:=16Ø;▪
 END;▪
 END;▪
{-----------------------------------}▪
PROCEDURE SOU;▪
 BEGIN▪
 FOR J:=125 TO 565 DO▪
 BEGIN▪
 SOUND (J);DELAY(1Ø);▪
 END;▪
 NOSOUND;▪
 END;▪
{-----------------------------------}▪
PROCEDURE CHOC;▪
 BEGIN▪
 FOR J:=3ØØ TO 5ØØ DO BEGIN▪
 SOUND (J);DELAY(5);END;NOSOUND;▪
 SETCOLOR(Ø);▪
 LINE(1,165,XX2+O,2Ø);▪
 LINE(319,165,X1+O,2Ø);▪
 O1:=Ø;O:=Ø;XX2:=13Ø;X1:=19Ø;▪
 DEC(SC1);▪
 SETCOLOR(2);▪
 STR(SC1,SC1_);▪
 SETFILLSTYLE(1,1);▪
 FILLELLIPSE(194,182,12,7);▪
 OUTTEXTXY(194,182,SC1_);▪
 END;▪
{-----------------------------------}▪
PROCEDURE CHANGEPOS;▪
BEGIN▪
CASE B OF▪
 2:BEGIN INC(O,6);END;▪
 3:BEGIN DEC(O,6);END;▪
 END;▪
 IF (X1+O>175) AND (B=1) THEN INC(O,4);▪
 IF (XX2+O<145) AND (B=1) THEN DEC(O,4);
▪
 INC(I2);▪
 IF I2>1ØØ THEN BEGIN▪
 RANDOMIZE;▪
 I2:=Ø;▪
 O1:=RANDOM(3);▪
 CASE O1 OF▪
 Ø:O1:=-5;▪
 1:O1:=Ø;▪
 2:O1:=5;▪
 END;▪
 END;▪
 O:=O+O1;▪
END;▪
{-----------------------------------}▪
PROCEDURE PERDU;▪
 BEGIN▪
 CLEARDEVICE;▪
 SETCOLOR(2);▪
 SETTEXTSTYLE(4,Ø,8);▪
 RECTANGLE(1,1,319,199);▪
 SETCOLOR(1);▪
 OUTTEXTXY(16Ø,5Ø,'FINI');▪
 SETCOLOR(3);▪
 SETTEXTSTYLE(4,Ø,5);▪
 OUTTEXTXY(16Ø,13Ø,NOM);▪
 SOUND(25Ø);DELAY(1ØØØ);NOSOUND;SOUND(15
Ø);DELAY(1ØØØ);NOSOUND;▪
 SOUND(25Ø);DELAY(1ØØØ);NOSOUND;SOUND(15
Ø);DELAY(1ØØØ);NOSOUND;▪
 SC1:=Ø;▪
 END;▪
{-----------------------------------}▪
PROCEDURE ENNEMI;▪
 BEGIN▪
 SETFILLSTYLE(Ø,Ø);▪
 O3:=O/112;▪
 CASE BO OF▪
 1:O4:=155+O;▪
 2:O4:=ROUND(145+(83*O3));▪
 3:O4:=ROUND(115+(63*O3));▪
 4:O4:=ROUND(85+(43*O3));▪
 END;▪
 O5:=11;▪
 CASE BO OF▪
 1:PUTIMAGE(O4,O5,PF1^,1);▪
 2:BEGIN BAR(O2,11,O2+15,18);O5:=5Ø;PUTI
MAGE(O4,O5,PF2^,1);END;▪
 3:BEGIN BAR(O2,5Ø,O2+35,71);O5:=7Ø;PUTI
MAGE(O4,O5,PF3^,1);END;▪
 4:BEGIN BAR(O2,7Ø,O2+95,133);O5:=9Ø;PUT
IMAGE(O4,O5,PF4^,1);SOU;▪
 SC1:=Ø;PUTIMAGE(O4,O5,PF4^,1);END;▪
 END;▪
 O2:=O4;▪
 END;▪
{-----------------------------------}▪
PROCEDURE ENNEMI1;▪
 BEGIN▪
 SETFILLSTYLE(Ø,Ø);▪
 CASE BO OF▪
 1:BAR(O2,3,O2+15,18);▪
 2:BAR(O2,43,O2+35,71);▪
 3:BAR(O2,64,O2+95,133);▪
 END;▪
 TOUCHE:=#Ø;▪
 END;▪
{-----------------------------------}▪
PROCEDURE TIR;▪
VAR O6,O7:INTEGER;▪
 BEGIN▪
 ROUTE1;▪
 SETCOLOR(3);▪
 LINE(135,141,155+O,21);LINE(185,141,165
+O,21);▪
 CASE O5 OF▪
 11:BEGIN O6:=O4+5;O7:=O5-4;END;▪
 5Ø:BEGIN O6:=O4+15;O7:=O5-4;END;▪
 7Ø:BEGIN O6:=O4+45;O7:=O5-4;END;▪
 END;▪
 IF SC2=1 THEN BEGIN SETCOLOR(1);OUTTEXT
```

```
XY(O6,O7,'PAF');END;▪
 FOR J:=600 TO 700 DO BEGIN▪
 SOUND(J);DELAY(5);▪
 END;▪
 NOSOUND;▪
 IF (SC2=1) THEN BEGIN I:=320;SC2:=0;ENN
EMI1;END;▪
 SETCOLOR(0);▪
 LINE(135,141,155+O,21);LINE(185,141,165
+O,21);▪
 TOUCHE:=#0;▪
 END;▪
{-----------------------------------}▪
PROCEDURE COMENNEMIS;▪
 BEGIN▪
 {---SYSTEME ACCELERATEUR DES ENNEMIS---
}▪
 INC(KL1);KL3:=FORCE-KL;▪
 IF KL3<2 THEN KL3:=2;▪
 IF KL1>=KL3 THEN BEGIN KL1:=0;INC(T1);K
L2:=1;END;▪
 IF KL2=1 THEN BEGIN▪
 CASE T1 OF▪
 1:BEGIN BO:=1;ENNEMI;END;▪
 2:BEGIN BO:=2;ENNEMI;END;▪
 3:BEGIN BO:=3;ENNEMI;END;▪
 4:BEGIN BO:=4;ENNEMI;END;▪
 END;▪
 KL2:=0;▪
 END;▪
 END;▪
{-----------------------------------}▪
PROCEDURE MOTEUR;▪
 BEGIN▪
 CASE JOY1 OF▪
 1: STICK(XX,YY,B1,B2);▪
 2: SOUR(B1,B2,XX,YY);▪
 END;▪
 WHILE (TOUCHE<>#27) AND (SC1>0) DO▪
 BEGIN▪
 TOUCHE:=UPCASE(READKEY);▪
 WHILE (NOT KEYPRESSED) AND (SC1>0) DO▪
 BEGIN▪
 IF YY>190 THEN I:=320;▪
 IF (B1=1) OR (TOUCHE='7') OR (B2=1) OR
(TOUCHE='9') THEN TIR;▪
 {---COMPTEUR KILOMETRIQUE---}▪
 IF I<250 THEN INC(SC);▪
 IF SC=100 THEN▪
 BEGIN▪
 SETCOLOR(2);▪
 INC(KL);▪
 STR(KL,KL_);▪
 SETFILLSTYLE(1,1);▪
 FILLELLIPSE(126,182,12,7);▪
 OUTTEXTXY(126,182,KL_);▪
 SC:=0;SC2:=1;T1:=0;▪
 END;▪
 SCT:=10-SC1;▪
 ROUTE1;▪
 {---CHOIX DES SYSTEMES---}▪
 CASE JOY1 OF▪
 1: STICK(XX,YY,B1,B2);▪
 2: SOUR(B1,B2,XX,YY);▪
 3: FLECHES;▪
 END;▪
 {---POSITIONS DES VOLANTS---}▪
 CASE JOY1 OF▪
 1:BEGIN▪
 IF (YY<20)THEN ZA:=TRUE ELSE ZA:=FALSE;
▪
 IF (XX>180) AND (B=1) THEN VOLANTD;▪
 IF (XX<10) AND (B=1) THEN VOLANTG;▪
 IF (XX>20) AND (XX<170) AND (B<>1) THEN
 VOLANTC;END;▪
 2:BEGIN▪
 IF (YY<80)THEN ZA:=TRUE ELSE ZA:=FALSE;
▪
 IF (XX>180) AND (B=1) THEN VOLANTD;▪
 IF (XX<140) AND (B=1) THEN VOLANTG;▪
 IF (XX>140) AND (XX<180) AND (B<>1) THE
N VOLANTC;END;▪
 3:BEGIN▪
 IF (YY<20)THEN ZA:=TRUE ELSE ZA:=FALSE;
▪
 IF (XX>290) AND (B=1) THEN VOLANTD;▪
 IF (XX<10) AND (B=1) THEN VOLANTG;▪
 IF (XX>20) AND (XX<280) AND (B<>1) THEN
 VOLANTC;END;▪
 END;▪
 {---DESSIN DU TACHIMETRE---}▪
 SETCOLOR(0);▪
 ARC(X,Y,0,I,A-5);▪
 GETARCCOORDS(CO);▪
 SETCOLOR(1);▪
 SETLINESTYLE(0,0,0);▪
 LINE(CO.X,CO.Y,CO.XEND,CO.YEND);▪
 TACHI;▪
 IF (GETPIXEL(34,149)=1) OR (GETPIXEL(28
5,149)=1) THEN CHOC;▪
 ROUTE0;▪
 IF SC2=0 THEN CHANGEPOS;▪
 IF (SC2=1) AND (I<250) THEN COMENNEMIS;
▪
 END;▪
 END;▪
 END;▪
{-------PROGRAMME PRINCIPAL--------}▪
 BEGIN▪
 CLRSCR;▪
 ENTREE;▪
 ZE:=TRUE;COMPT:=0;MUST:=0;▪
 FOR I:=1 TO 10 DO BEGIN SCORT[I]:=0;END
;▪
 REPEAT▪
 BEGIN▪
 {---INITIALISATION DES VARIABLES---}▪
 INITIALISATION;▪
 {---INITIALISATION DU MODE---}▪
 INITGRAPH(DRI,MODE,'');▪
 {---DESSIN DES MOBILES---}▪
 IF ZE THEN DESSIN_MOB;▪
 DESSIN_ECRAN;▪
 MOTEUR;▪
 IF SC1=0 THEN PERDU;▪
 CLOSEGRAPH;▪
 SORTIE;▪
 END;▪
 UNTIL (TOUCHE='N') OR (TOUCHE='n');▪
 CLRSCR;▪
 END.{---FIN DE PROGRAMME---}▪
▪
```